ISO/IEC JTC 1/SC 34/JWG 7

Joint JTC 1/SC 34-TC 46/SC 4-IEC/TC 100/TA 10 WG: EPUB

Convenorship: KATS (Korea, Republic of)

Document type:	Text for CD ballot or comment
Title:	Text of ISO/IEC PDTS2 22424-2 Preserving Content in EPUB Format - Part 2: Metadata requirements
Status:	This document is prepared by project editor, Juha Hakala, to reflect all comments from the CRM of ISO/IEC PDTS 22424-2 Preserving Content in EPUB Format - Part 2: Metadata requirements
Date of document:	2018-07-24
Source:	Juha Hakala (project editor)
Expected action:	INFO
No. of pages:	54
Email of convenor:	samoh21@gmail.com, zzosang@gmail.com
Committee URL:	https://isotc.iso.org/livelink/livelink/open/jtc1sc34jwg7

ISO/IEC DTS 22424-2:2018

ISO/IEC JTC 1/SC 34/JWG 7

Secretariat: JISC

Information technology — Digital publishing — Preserving Content in EPUB Format - Part 2: Metadata requirements

Technologies de l'information - Édition numérique - Archivage pérenne de l'EPUB3 – Partie II : Exigences sur les métadonnées

DTS stage

Warning for WDs and CDs

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Contents

49 # Foreword

50 ISO (the International Organization for Standardization) is a worldwide federation of national
51 standards bodies (ISO member bodies). The work of preparing International Standards is normally
52 carried out through ISO technical committees. Each member body interested in a subject for which a
53 technical committee has been established has the right to be represented on that committee.
54 International organizations, governmental and non-governmental, in liaison with ISO, also take part in
55 the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all
56 matters of electrotechnical standardization.

57 The procedures used to develop this document and those intended for its further maintenance are
58 described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the
59 different types of ISO documents should be noted. This document was drafted in accordance with the
60 editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

61 Attention is drawn to the possibility that some of the elements of this document may be the subject of
62 patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of
63 any patent rights identified during the development of the document will be in the Introduction and/or
64 on the ISO list of patent declarations received (see www.iso.org/patents).

65 Any trade name used in this document is information given for the convenience of users and does not
66 constitute an endorsement.

67 For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and
68 expressions related to conformity assessment, as well as information about ISO's adherence to the
69 World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following
70 URL: www.iso.org/iso/foreword.html.

71 This document was prepared by Technical Committee ISO/IEC JTC 1, Joint Technical Committee,
72 Subcommittee SC 34, Document Description and Processing Language.

73 ISO/IEC DTS 22424 consists of the following parts, under the general title *Information technology —*
74 *Digital publishing — EPUB 3 Preservation*:

75 — *Part 1: Principles*

76 — *Part 2: Metadata requirements*

77

Introduction

This document facilitates the long-term preservation of EPUB publications by specifying metadata elements which are required or recommended for long-term preservation (such as identifiers) and the ways in which the EPUB publication and related metadata can be packaged. EPUB versions 3.0 and later are covered; if necessary, the EPUB version applicable is specified.

Long term preservation in general requires two things:

- making the object such as EPUB publication fit for preservation – including features to be used and feature to avoid;

- the packaging of the object (and any metadata related to it) together with any additional data such as other versions of the object and other documentation into an OAIS Submission Information Package (SIP).

Part 1 of this technical specification concentrates on the archivability of EPUB documents.

The background to this document comes from the Open Archival Information System, which is described in Part 1 of this standard.

When a Submission Information Package (SIP) is formed, mandatory preservation metadata SHALL be present in the package. Depending on the agreements made between the producer and the archive, metadata elements are stored either in the container document or the EPUB publication itself, or both. Usually an archive would expect to find all relevant metadata in the container, unless the submission agreement allows embedding of metadata into EPUB publications.

This document does not require any changes to be made to the current of future EPUB standards. However, when an EPUB publication is created or modified for submission to an archive, there are some EPUB features that should be used and others that should be avoided. Part 1 of this technical specification describes how the EPUB format SHOULD be applied. This Part 2 document concentrates on mandatory and recommended metadata elements needed for the long-term preservation of EPUB publications and their METS encoding. Part 1 recommends the usage of METS but allows also other container standards; this part concentrates on preservation metadata and its METS encoding in SIPs. Future versions of this standards MAY specify other encodings.

In order to guarantee access to documents, OAIS archives may migrate documents into new file formats when the original formats are no longer supported by commonly used rendering tools. If the document to be migrated is an e-book in an outdated EPUB format, migration can be made to a more modern version of EPUB or, at least in principle, to another e-book format. Migration into a future version of EPUB is easy, if successive EPUB versions are compatible. Migration to other formats may not be, since as of this writing some EPUB features are not supported in other e-book formats, and vice versa. And even if the same feature is supported, technical implementations can be incompatible. For instance, if an EPUB 3 publication using fixed layout is migrated to Amazon's KF8 format, preserving fixed layout properties would require special attention since there are significant technical differences between formats in this respect.

Depending on source and target formats, migration will not always produce a satisfactory digital copy of an e-book. Sometimes migration cannot be applied at all; programs cannot be migrated without access to and good understanding of the source code. In such cases long term preservation is possible only if the OAIS archive responsible is able to emulate either the program's original hardware or software environment.

120 Within the preservation community, emulation is considered to be a viable option for some content. For
121 the time being there is no full understanding on how emulation will function in the long term, but this
122 may change with Emulation as a Service approaches coming to the market.

123 Metadata requirements in this document are based on the migration of file formats. Emulation is not
124 covered (just a single example of emulation-related preservation metadata is given), although in the
125 EPUB context, interactive publications containing e.g. JavaScript may have to be preserved with
126 emulation. Detailed preservation metadata requirements for emulation-based preservation strategy
127 may be added into a future version of this document.

128 Supporting emulation might require just information about appropriate tools in the submission
129 agreement or in the related documentation. A more sustainable approach is to include a description of
130 the emulation environment (hardware and/or software) in the premis:object section of the PREMIS
131 metadata record in the SIP. During ingest this information is copied into AIP. If migration is used,
132 hardware and software environments needed for rendering the versions of the document in the AIP can
133 be specified separately as access environments.

134 Since the purpose of a migration is to preserve the intellectual content, it is possible that the look and
135 feel of preserved documents will eventually be lost. This may become a problem for fixed layout
136 documents including EPUB publications using this feature. If semantics and layout are interlinked, it
137 may be necessary to archive the original EPUB publication even after the migration in order to
138 guarantee access to the original content.

139 Migration both requires and produces preservation metadata. For instance, staff in the archives has to
140 figure out which tools can be used to carry out the migration, and what weak points they may have. The
141 intention of the preservation community is to maintain this information in format libraries such as
142 PRONOM[1] . When a new Archival Information Package (AIP) is created after a migration, the package
143 should contain both the old and the new representation of the migrated document and preservation
144 metadata describing the migration event and the possible differences between the document versions[2].
145 Depending on their needs and archived resources archive users can then make a choice between the
146 original, which is authentic but possibly difficult to render, and the migrated document, which should
147 be easy to use but less authentic. In practice, finding access software to outdated versions of preserved
148 documents may be difficult. The OAIS archive, on the other hand, can migrate the original document
149 again when better tools can be used, or if there are significant issues in migrated documents.

150 Metadata elements that SHALL be included in SIPs are a priori essential for digital preservation. For
151 instance, if there is no digital signature present and a secure transfer channel has not been used, it is
152 impossible to guarantee the information entering the archive has not changed during transfer or that it
153 is coming from a correct source. Moreover, if the data has already been tampered with before it enters
154 the archive, all subsequent preservation actions may be useless.

155 This specification does not specify generic conformance requirements for EPUB publications, but may
156 make some restrictions to the use of EPUB specifications. The generic conformance requirements made
157 in the EPUB Contents Documents Specification apply to EPUB publications in SIPs as well.

158 Part 1 of this document defined a set of requirements for archivable EPUB publications. Below is a short
159 summary of these requirements:

160 • SIPs SHALL contain the entire EPUB publication including the fonts used. All publication
161 resources SHOULD be embedded in the EPUB container, including audio and video resources.

---

[1] http://www.nationalarchives.gov.uk/PRONOM/Default.aspx

[2] This specification is only concerned with those metadata elements which are to be included in SIPs. Preservation
metadata needed in AIPs (describing for instance the impact of migration on the document) is beyond the scope of
this specification.

162   Linked resources MAY be used if the archive is able to retrieve the resources during ingest and
163   incorporate them into the AIP. SIPs SHALL NOT contain viruses or anything else not part of the
164   submitted EPUB publications.

165   • Preview EPUB publications MAY be submitted, if it will be replaced with the final version of the
166   publication once it is available, or if it is not possible to submit the final version.

167   • Submitted resources SHOULD NOT be DRM protected, encrypted, or obfuscated. If any of these
168   mechanisms have been used, the archive SHALL be permitted to remove them during ingest.

169   • If an EPUB content document in a SIP contains scripting, the EPUB publication SHALL contain a
170   fallback for the content in question. In the EPUB context scripting enables the use of JavaScript
171   applications for e.g. image manipulation or enabling dynamic changes of the content. Preserving
172   such functionality in the long term can be difficult and might require emulation.

173   • If there are core media type resources or foreign resources in EPUB publications that need to be
174   preserved, their file formats SHOULD be approved for ingest and/or preservation[3]. If there is
175   un-archivable content, these resources SHALL be an archivable fallback and the files that cannot
176   be preserved (except in bit level) SHALL be encoded in a way that they SHALL NOT be checked
177   during ingest but stored as such in AIP.

178   • EPUB reader or readers that is/are known to be able to render the submitted publication
179   correctly SHOULD be specified in the SIP.

180   • Canonical fragment identifiers[4] SHOULD NOT be used in EPUB publications submitted to an
181   archive, because if/when the publication is migrated into another file format, these identifiers
182   may stop functioning properly.

183   • Fixed layout documents SHOULD be modified before submission so that the meaning is not
184   dependent on the layout and therefore preserving the original look and feel will not be
185   necessary.

186 For a more complete description of the above requirements, please consult Part 1 of this document.

187

---

[3] Acceptable formats are defined in the submission agreement.

[4] http://www.idpf.org/epub/linking/cfi/epub-cfi.html

Information technology — Digital publishing — Preserving Content in EPUB Format - Part 2: Metadata requirements

1   Scope

This document supports long term preservation of EPUB publications via a dual strategy. First, it considers EPUB features from long term preservation point. Some EPUB features are forbidden and some others required, depending on they relate to long term preservation. An EPUB document constructed according to these guidelines are suitable for preservation. In this respect, this specification is related to EPUB in the same way than PDF/A is related to PDF.

Second, this specification makes EPUB compliant with current practices of digital archives and technical requirements of long term preservation applications. The former tend to rely on Open Archival Information Systems (OAIS) in their operations; the latter prefer to ingest electronic documents only in containers which conform to standards such as METS (Metadata Encoding and Transmission Standard).

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.

ISO/IEC TS 30135, *Information technology — Digital publishing — EPUB3*

ISO 14721. *Space data and information transfer systems – Open archival information system (OAIS) – Reference model*

ISO 15836-1:2017. *Information and documentation –The Dublin Core metadata element set – Part 1: Core elements.*

METS. *Metadata Encoding & Transmission Standard. Version 1.11.* [online]. Library of Congress, 2015. Available from: https://www.loc.gov/standards/mets/

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

3   Terms and definitions

For the purposes of this document, the following terms and definitions apply. Unless stated otherwise, the terms have been adopted from ISO 14721:2012.

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

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

3.1
access functional entity
OAIS functional entity that contains the services and functions, which make the archival information holdings and related services visible to Consumers

3.2
administrative metadata
metadata that provides information to help manage a resource, such as when and how it was created, file type and other technical information, and access rights

[SOURCE: Understanding metadata]

3.3
archival information package
AIP
Information Package consisting of Content Information and associated Preservation Description Information (PDI), which is preserved within an OAIS

3.4
archive
OAIS archive
organization that intends to preserve information for access and use by a Designated Community

3.5
authenticity
property than an entity is what it claims to be

[SOURCE: ISO/IEC 27000]

Note 1 to entry: Authenticity is judged on the basis of evidence.

3.6
bit preservation
term used to denote a very basic level of preservation of digital resource as it has been submitted (literally the preservation of the **bits** forming a digital resource)

Note 1 to entry: This may include maintaining onsite and offsite backup copies, virus checking, fixity-checking, and periodic refreshing to a new storage medium.

Note 2 to entry: Bit preservation is not digital preservation but it does provide a building block for the more complete set of digital preservation practices and processes that ensure the survival of digital content and also its usability, display, context and interpretation over time.

[SOURCE: Digital preservation handbook, Glossary]

3.7
consumer
role played by those persons or client systems, who interact with OAIS services to find preserved information of interest and to access that information in detail

Note 1 to entry: This can include other OAISs, as well as internal OAIS persons or systems.

3.8
content information
set of information that is the original target of preservation or that includes part or all of that information

Note 1 to entry: It is an Information Object composed of its Content Data Object and its Representation Information.

264 **3.9**
265 **context information**
266 information that documents the relationships of the Content Information to its environment

267 Note 1 to entry: This includes reasons why the Content Information was created and how it relates to other
268 Content Information objects.

269 **3.10**
270 **core media type**
271 a set of publication resource for which no fallback is required.

272 [SOURCE: EPUB Publications 3.0 Recommended Specification 11 October 2011]

273 Note 1 to entry: Core media types have been specified in chapter 5.1. of the EPUB publications specification,
274 version 3.0.1.

275 EXAMPLE   core media types for still images are image/gif, image/jpg, image/png and image/svg+xml. Any
276   other still image file format is foreign and requires a fallback, meaning the same resource
277   expressed in another foreign format or core media type.

278 **3.11**
279 **data, pl**
280 reinterpretable representation of information in a formalized manner suitable for communication,
281 interpretation, or processing

282 [SOURCE: ISO 5127:2017]

283 Note 1 to entry: Data are often understood as taking the form of a set of values of qualitative or quantitative
284 variables.

285 **3.12**
286 **data dictionary**
287 organized and constructed (electronic data base) compilation of descriptions of data concepts that
288 provides a consistent means for documenting, storing and retrieving the syntactical form (i.e.
289 representational form) and the meaning and connotation of each data concept

290 [SOURCE: ISO 24531:2013]

291 Note 1 to entry: PREMIS[5] is a data dictionary.

292 **3.13**
293 **descriptive metadata**
294 **descriptive information**
295 metadata about a resource for example for discovery and identification

296 Note 1 to entry: These can include elements such as title, abstract, author, and keywords.

297 [SOURCE: Understanding metadata]

298 **3.14**
299 **designated community**
300 identified group of potential Consumers who should be able to understand a particular set of
301 information

---

[5] PREMIS Data Dictionary for Preservation Metadata (https://www.loc.gov/standards/premis/) is a leading
metadata specification for metadata needed for long-term preservation.

3

302 Note 1 to entry: A Designated Community may be composed of multiple user communities. The community is
303 defined by an Archive, though this definition may change later on.

304 **3.15**
305 **digital preservation**
306 series of managed activities necessary to ensure continued access to digital materials for as long as
307 necessary

308 Note 1 to entry: Digital preservation refers to all of the actions required to maintain access to digital materials
309 beyond the limits of media failure or technological and organizational change

310 Note 2 to entry: Those materials may be records created during the day-to-day business of an organization; "born-
311 digital" materials created for a specific purpose (e.g. teaching resources); or the products of digitisation projects.

312 EXAMPLE 1   **Short-term preservation** - Access to digital materials either for a defined period of time while
313   use is predicted but which does not extend beyond the foreseeable future and/or until it becomes
314   inaccessible because of changes in technology.

315 EXAMPLE 2   **Medium-term preservation** - Continued access to digital materials beyond changes in
316   technology for a defined period of time but not indefinitely.

317 EXAMPLE 3   **Long-term preservation** - Continued access to digital materials, or at least to the information
318   contained in them, indefinitely.

319 [SOURCE: Digital preservation handbook, Glossary]

320 **3.16**
321 **digital rights management**
322 **DRM**
323 packaging, distributing, controlling, and tracking content based on rights and licensing information

324 [SOURCE: ISO 19153:2014]

325 **3.17**
326 **digital signature**
327 **signature**
328 data appended to, or a cryptographic transformation of, a data unit that allows the recipient of the data
329 unit to prove the source and integrity of the data unit and protect against forgery, e.g. by the recipient

330 [SOURCE: ISO/IEC 19784-1:2006]

331 **3.18**
332 **dissemination information package**
333 **DIP**
334 information package, derived from one or more AIPs, sent by an Archive to a Consumer in response to a
335 request in the OAIS

336 **3.19**
337 **distributable object**
338 component of an EPUB publication that can be reused in other contexts

339 Note 1 to entry: A Distributable Object can be a complete EPUB Content Document (e.g., a chapter of a book), a
340 section of such a document (e.g., an exercise or a promotional excerpt), a media resource (e.g., a video or
341 interactive feature), or a combination of such resources that are not necessarily contiguous within the parent
342 EPUB publication but are intended to be able to be distributed as a unit.

343 [SOURCE: EPUB Distributable Objects 1.0]

3.20
DRM
digital rights management
packaging, distributing, controlling, and tracking content based on rights and licensing information
[SOURCE: ISO 19153:2014]

3.21
electronic book
e-book
non-serial digital document, licensed or not, where searchable text is prevalent, and which can be seen in analogy to a print book

Note 1 to entry: The use of e-books is, in many cases, dependent on a dedicated device and/or a special reader or viewing software.

[SOURCE: ISO 2789:2013]

3.22
EPUB container
ZIP based packaging and distribution format for EPUB publications

[SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

3.23
EPUB content document
publication resource that conforms to one of the EPUB content document definitions

[SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

3.24
EPUB navigation document
specialization of the XHTML content document, containing human- and machine-readable global navigation information

[SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

3.25
EPUB publication
collection of one or more renditions conforming to the EPUB specifications, packaged in an EPUB container

[SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

3.26
EPUB reading system
system that processes EPUB publications for presentation to a user in a manner compliant with EPUB specifications

[SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

3.27
fallback
mechanism with which versions of the same resource in different file formats can be linked to one another

[SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

385 Note 1 to entry: A reading system that does not support the file format of a foreign resource shall traverse the
386 fallback chain until it finds a version it can render.

387 **3.28**
388 **fixity information**
389 information that documents the authentication mechanisms and provides authentication keys to ensure
390 that the Content Information object has not been altered in an undocumented manner

391 [SOURCE: ISO 13527:2010]

392 **3.29**
393 **foreign resource**
394 publication resource that is not a core media type

395 [SOURCE: EPUB Publications 3.0 Recommended Specification 11 October 2011]

396 Note 1 to entry: According to EPUB 3.1, foreign resources require at least one fallback if they are in the spine or
397 embedded in EPUB Content Documents.

398 **3.30**
399 **identifier**
400 data string or pointer that establishes the identity of an item, institution, or person alone or in
401 combination with other elements.

402 [SOURCE: ISO 8459:2009]

403 Note 1 to entry: EPUB 3 specifies Unique Identifiers and Release Identifiers; the latter is a combination of a Unique
404 Identifier and the last modification data of the rendition of the resource.

405 **3.31**
406 **independently understandable**
407 characteristic of information that is sufficiently complete to allow it to be interpreted, understood, and
408 used by the Designated Community without having to resort to special resources not widely available,
409 including named individuals

410 **3.32**
411 **information**
412 any type of knowledge that can be exchanged

413 Note 1 to entry: In an exchange, this is represented by data

414 EXAMPLE   a string of bits (the data) accompanied by a description on how to interpret the string of
415   bits as numbers representing temperature observations measured in degrees Celsius
416   (the representation information)

417 **3.33**
418 **information package**
419 logical container composed of optional content information and optional associated preservation
420 description information

421 **3.34**
422 **ingest functional entity**
423 OAIS functional entity that contains the services and functions that accept SIPs from producers,
424 prepares AIPs for storage, and ensures AIPs and their supporting descriptive information become
425 established within the OAIS

426 **3.35**
427 **long-term**
428 period of time long enough to raise concerns about the impact of changing technologies, including
429 support for new media and data formats, and of a changing designated community, on the information
430 being held in an OAIS

431 Note 1 to entry: This period extends into the indefinite future.

432 **3.36**
433 **long-term preservation**
434 act of maintaining information, independently understandable by a designated community, with
435 evidence supporting its authenticity over the long term

436 **3.37**
437 **manifest**
438 EPUB manifest element provides an exhaustive list of the Publication Resources that constitute the
439 given Rendition, each represented by an item element.

440 [SOURCE: EPUB Publications 3.0.1]

441 **3.38**
442 **metadata**
443 data about other Data, documents, or records that describe their content, context, structure, format,
444 provenance, and/or rights.

445 [SOURCE: ISO 5127:2017]

446 **3.39**
447 **METS**
448 Metadata Encoding and Transmission Standard, a standard for presenting metadata using XML.

449 [SOURCE: Digital preservation handbook, Glossary]

450 **3.40**
451 **migration**
452 means of overcoming technological obsolescence by transferring digital resources from one
453 hardware/software generation to the next

454 Note 1 to entry: The purpose of migration is to preserve the intellectual content of digital objects and to retain the
455 ability for clients to retrieve, display, and otherwise use them in the face of constantly changing technology.

456 Note 2 to entry: Migration differs from the refreshing of storage media in that it is not always possible to make an
457 exact digital copy or replicate original features and appearance and still maintain the compatibility of the resource
458 with the new generation of technology.

459 [SOURCE: Digital preservation handbook, Glossary]

460 **3.41**
461 **Open Archival Information System**
462 **OAIS**
463 archive, consisting of an organization, which may be a part of a larger organization, of people and
464 systems, that has accepted the responsibility to preserve Information and make it available to a
465 Designated Community. It has a set of responsibilities, as defined in section 4, which allow an OAIS
466 Archive to be distinguished from other uses of the term 'Archive'.

467 Note 1 to entry: The term 'Open' in OAIS is used to imply that this Recommendation and future related
468 Recommendations and standards are developed in open forums, but it does not imply access to the Archive is
469 unrestricted.

470 Note 2 to entry: The OAIS abbreviation is also commonly used to refer to the Open Archival Information System
471 Reference Model standard which defined the term. The standard is a conceptual framework describing the
472 environment, functional components, and information objects associated with a system responsible for long-term
473 preservation.

474 **3.42**
475 **package document**
476 publication resource that describes one rendition of an EPUB publication, as defined in package
477 document. The package document carries meta information about the Rendition, provides a manifest of
478 resources and defines the default reading order.

479 [SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

480 Note 1 to entry: It specifies all tools required to render the document, provides an exhaustive list of resources
481 belonging to the document, and defines their default reading order.

482 **3.43**
483 **PDF**
484 Portable Document Format, a set of formats and open standards maintained by the International
485 Organization for Standardization for producing and sharing electronic documents

486 Note 1 to entry: Originally developed by Adobe Systems.

487 [SOURCE: Digital preservation handbook, Glossary]

488 **3.44**
489 **PDF/A**
490 versions of the PDF standard intended for archival use

491 [SOURCE: Digital preservation handbook, Glossary]

492 **3.45**
493 **pre-ingest**
494 actions required before data can be submitted into an OAIS archive, including negotiation of data
495 acquisitions, checking rights and access criteria, licensing, and data submission

496 Note 1 to entry: This area also includes activities involving data producer support and training.

497 Note 2 to entry: Pre-ingest is not a function in the standard OAIS model, but activities in this area can form a
498 significant part of a producer's responsibilities.

499 [SOURCE: UK Data Archive. Archive training manual6]

500 **3.46**
501 **preservation description information**
502 **PDI**
503 information necessary for the adequate preservation of Content Information that can be categorized as
504 provenance, reference, fixity, context, and rights information

---

6 http://www.data-archive.ac.uk/curate/archive-training-manual/pre-ingest

505 **3.47**
506 **preservation metadata**
507 metadata containing information needed to archive and preserve a resource

508 [SOURCE: Understanding metadata]

509 **3.48**
510 **preservation planning functional entity**
511 OAIS functional entity that provides the services and functions for monitoring the environment of the
512 OAIS and that provides recommendations and preservation plans to ensure information stored in the
513 OAIS remains accessible to, understandable by, and sufficiently usable by the designated community
514 over the long term, even if the original computing environment becomes obsolete

515 **3.49**
516 **producer**
517 role played by those persons or client systems that provide the information to be preserved

518 Note 1 to entry: This can include other OAISs or internal OAIS persons or systems.

519 **3.50**
520 **provenance information**
521 information that documents the history of the Content Information

522 Note 1 to entry: This information states the origin or source of the Content Information, any changes that may
523 have taken place since it was generated, and who has had custody of it.

524 Note 2 to entry: The Archive is responsible for creating and preserving Provenance Information from the point of
525 ingest; however, earlier Provenance Information should be provided by the Producer. Provenance Information
526 adds to the evidence to support authenticity.

527 **3.51**
528 **publication resource**
529 resource that has the content or instructions contributing to the logic and rendering of at least one
530 rendition of an EPUB publication

531 EXAMPLE   Examples of publication resources include a rendition's Package Document, EPUB
532   Content Document, EPUB style sheets, audio, video, images, and embedded fonts and
533   scripts.

534 **3.52**
535 **reading system**
536 system that processes EPUB publications for presentation to a user in a manner conformant with EPUB
537 specification

538 [SOURCE: Modified from EPUB 3.1 Recommended Specification 5 January 2017].

539 **3.53**
540 **reference information**
541 information that is used as an Identifier for the Content Information

542 Note 1 to entry: This also includes Identifiers that allow outside systems to refer unambiguously to a particular
543 Content Information.

544 EXAMPLE   an ISBN is a type of Reference Information.

545 **3.54**
546 **reference model**
547 framework for understanding significant relationships among entities in an environment and for the
548 development of consistent standards or specifications supporting that environment

549 Note 1 to entry: A Reference Model is based on a small number of unifying concepts and may be used as a basis for
550 education and explaining standards to a non-specialist.

551 **3.55**
552 **reformatting**
553 copying information content from one storage medium to a different storage medium (media
554 reformatting) or converting from one file format to a different file format (file reformatting)

555 [SOURCE: Digital preservation handbook, Glossary]

556 **3.56**
557 **refreshing**
558 copying information content from one storage media to the same storage media

559 [SOURCE: Digital preservation handbook, Glossary]

560 **3.57**
561 **release identifier**
562 identifier that allows any instance of an EPUB publication to be compared against another to determine
563 if they are identical, different versions, or unrelated

564 [SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

565 Note 1 to entry: Release Identifiers consist of a unique identifier and the last-modified date of the document.

566 **3.58**
567 **remotely-hosted resource**
568 objects hosted outside the EPUB Container.

569 Note 1 to entry: EPUB 3.1 allows fonts and resources used by scripts to be hosted externally.

570 **3.59**
571 **rendition**
572 one rendering of the content of an EPUB publication, as expressed by an EPUB package

573 [SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

574 **3.60**
575 **repository system**
576 long term preservation system used by an archive
577
578 **3.61**
579 **rights management metadata**
580 information that identifies the access restrictions concerning the Content Information, including the
581 legal framework, licensing terms, and access control

582 Note 1 to entry: This contains the access and distribution conditions stated in the Submission Agreement, related
583 to both preservation (by the OAIS) and final usage (by the Consumer).

584 Note 2 to entry: It also includes specifications for the application of rights enforcement measures.

585 **3.62**
586 **spine**
587 EPUB spine element defines the default reading order of the EPUB Publication content by defining an
588 ordered list of manifest item references.
589
590 [SOURCE : EPUB Publications 3.0.1]
591
592 **3.63**
593 **structural metadata**
594 metadata that indicates how compound objects are put together, for example how the pages of a
595 document are arranged to form chapters
596
596 [SOURCE: Understanding metadata]
597 **3.64**
598 **submission agreement**
599 agreement reached between an OAIS archive and a Producer that specifies a data model and any other
600 arrangements needed for the data submission session

601 Note 1 to entry: This data model identifies the format/content and the logical constructs used by the Producer and
602 how they are represented on each media delivery or in a telecommunication session.

603 **3.65**
604 **submission information package**
605 **SIP**
606 information package that is delivered by a Producer to an OAIS to be used to construct or update one or
607 more AIPs and/or the associated descriptive information.

608 **3.66**
609 **unique identifier**
610 primary identifier of an EPUB publication, which may be shared by one or several renditions of the
611 same EPUB publication that conform to the EPUB standard and embody the same content.

612 [SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

613 **3.67**
614 **XHTML content document**
615 EPUB content document that conforms to the profile for HTML defined in XHTML Content Documents

616 [SOURCE: EPUB 3.1 Recommended Specification 5 January 2017]

617 Note 1 to entry: see EPUB Content Documents 3.1, chapter 2.

618 # 4   Abbreviated terms

AIP   Archival Information Package

DIP   Dissemination Information Package

DRM   Digital Rights Management

OAIS   Open Archival Information System

PDI   Preservation Description Information

SIP   Submission Information Package

5 Syntax

This document provides examples of how metadata elements SHOULD be expressed using either

1) Metadata Encoding and Transmission Standard (METS[7]) version 1.11 and PREMIS Data Dictionary for Preservation Metadata (PREMIS[8]) version 2.2, and/or
2) EPUB version 3.x

for encoding SIPs. Other container standards MAY be used, but this document does not support their usage.

This dual approach was chosen because there are different options available for a producer to turn existing EPUB publications into SIPs:

1) All metadata (mandatory and otherwise) may be embedded in the EPUB publication.
2) Mandatory metadata is copied from EPUB document to the METS container if and when it is already present, or created and placed in the METS container (recommended approach).
3) Option 2, but a container standard other than METS is used.

The first option looks appealing because that way it would be relatively easy to create EPUB publications suitable for long-term preservation, especially if the mandatory metadata elements are already present (and if the EPUB publication itself does not have features unsuitable for preservation).

Unfortunately this approach has some issues:

- Commonly used repository systems expect information packages based on container standards such as METS. Current versions of these applications may not able to process SIPs which contain only an EPUB publication.
- Depending on the mandatory metadata required, it may not be possible to include all preservation metadata into EPUB publication.
- If there is no container document, it may be difficult to send multiple EPUB publications in a single SIP, or partial updates (for instance, only descriptive metadata about a publication that has already been archived.

Options 2 and 3 are based on the idea that there are two independent specifications, the core EPUB specification (currently version 3.1), and a container specification (this document). This allows the two communities (EPUB and digital archivists) to cooperate without putting unnecessary constraints on each other. Both specifications are independent from one another, which makes it easier and safer to manage them.

From a technical point of view, the main strength of the second option is that METS containers are almost universally accepted in long-term preservation applications. One reason for the popularity of the standard is that it is flexible – it is possible to embed any descriptive or administrative metadata into a METS document. Whatever mandatory metadata will be agreed upon by the producer and the OAIS archive, METS can be used as a container.

The option of using some other container standard than METS or EPUB has not been examined while preparing this document. ISO/IEC JTC 1/SC 34 JWG7 decided early on to use METS due to its technical features and popularity among long-term preservation application vendors as well as libraries, archives,

---

[7] http://www.loc.gov/standards/mets/

[8] http://www.loc.gov/standards/premis/

657 and museums. If and when other options emerge in the future, it is possible to extend this specification
658 to support other container standards as well.

659 Of the four options listed above the one described herein is number 3, the hybrid approach. Each SIP
660 SHALL have a METS document with mandatory descriptive and administrative metadata elements
661 embedded, using e.g. Dublin Core and PREMIS formats. The use of a separate, METS based preservation
662 layer enables the current long-term preservation applications to ingest EPUB publications. Producers
663 and OAIS archives MAY also choose other approaches, such as embedding all metadata in EPUB
664 publications or using another container standard. Whichever strategy is chosen, it should be planned
665 out carefully.

666 In the hybrid approach, some descriptive and administrative metadata needed during ingest MAY not
667 be copied to the METS document. In order to use this metadata, the OAIS archive SHALL have reading
668 systems or other applications which are able to render EPUB publications and extract the relevant
669 information from them.

670 This specification does not require copying of EPUB structural metadata to METS documents. Therefore
671 the structural metadata in METS is simple, only specifying the location of EPUB publication or
672 publications in the SIP but not their internal structure. EPUB reading systems would not be able to use
673 the structural metadata in a METS document, because they utilize structural metadata in the EPUB
674 spine element when publications are rendered.

675 In order to eliminate uncertainty concerning the syntax and semantics of SIPs, submission agreements
676 SHALL specify a METS profile or profiles which can be used to facilitate packaging of EPUB publications.
677 This specification can be used as a basis for these profiles. The profile can be part of the submission
678 agreement, or linked to it. The latter approach was chosen in the Finnish Digital Library initiative; the
679 benefit is that submission agreements will be relatively simple because technical details are stated in
680 the document "Metadata requirements and preparing content for digital preservation"[9]. Finnish Digital
681 Library initiative has published also a separate document titled "File formats"[10], which lists the file
682 formats suitable for ingest and preservation. Unfortunately this document does not contain guidelines
683 on how these file formats should be applied. EPUB is an example of a file format which is in principle
684 archivable, but in practice can be used in a way which may makes long term preservation challenging.
685 The purpose of Part 1 of this document is to provide guidelines for creation of archivable EPUB
686 publications.

687 Specifications, such as the ones created in Finnish Digital Library initiative, SHALL be sufficiently
688 detailed; for instance, they SHALL specify all mandatory metadata elements and all archivable or
689 ingestible file formats. Otherwise SIPs may lack crucial data, or contain files that cannot be processed.
690 Of course even this may not be sufficient; in addition to only saying that MXF, TIFF and EPUB are
691 archivable formats, it is also necessary to specify what type of MXF videos, TIFF images and EPUB
692 publications are acceptable. Digital archiving projects like the National Digital Library in Finland do not
693 necessarily have a mandate or resources for such work; that is why specifications like this one for EPUB,
694 AS-07 for archivable MXF[11] and TI/A[12] for archival of TIFF images are needed.

695 If just listing all the archivable file formats is not enough, it is also insufficient to provide a list of
696 mandatory preservation metadata elements. Element specific guidelines are often necessary. For
697 instance, it is not enough to just say that SIPs must contain identifiers for EPUB publications. Producer
698 and OAIS archive SHALL also agree on what needs to be identified (for instance, EPUB publications,
699 their component parts, metadata records), which identifiers (ISBNs, DOIs, URNs, etc.) are accepted and
700 – just to give an EPUB specific example on identifier usage – whether EPUB release identifiers are

---

[9] http://www.kdk.fi/images/tiedostot/NDL-Metadata_v1.6.1.pdf

[10] http://www.kdk.fi/images/tiedostot/NDL-File-Formats-v1.5.1-en.pdf

[11] https://www.amwa.tv/projects/AS-07.shtml

[12] http://ti-a.org/

701 acceptable. Metadata is crucial in digital archiving, because it affects all the steps in the preservation
702 process – ingest, archival, and dissemination. When a producer and an OAIS archive decide on which
703 identifiers to use, this may have an impact not only on SIPs, but also on Archival Information Packages
704 (AIPs) and Dissemination Information Packages (DIPs) the archive will be able to create.

705 ## 6   Packaging metadata

706 This chapter covers mainly metadata about the SIP (container) which is usually submitted using METS
707 elements and attributes.

708 NOTE   It is not possible to make a clear division between descriptive and administrative metadata. For
709 instance package creator information is normally just administrative metadata. But if the
710 package creator has modified the EPUB publication to make sure that SIP meets the
711 requirements of the submission agreement, the creator may have performed tasks which
712 normally belong to the editor of the publication. The name of the editor is regarded as
713 descriptive metadata.

714 ### 6.1   Package creator / submitter information

715 Both the name of the original creator of the package and the name of the submitting organization
716 SHALL be included in the METS header, if the submitting organization has made any changes to the
717 package. If the submitting organization has not modified the content, the creator name is sufficient.

718 If a secure transmission channel is used and it allows identification of the submitting organization,
719 submitter information MAY be omitted.

720 Creator / submitter identifier SHOULD be included, if the name alone does not uniquely identify the
721 organization. The identifier SHOULD be an ISNI or another standard identifier. The identifier system in
722 use SHALL be indicated.

723 **Examples**

724 SIP creator:

```
725 <mets:metsHdr CREATEDATE="2017-07-15T12:00:00" RECORDSTATUS="NEW">
726   <mets:agent ROLE="CREATOR" TYPE="ORGANIZATION">
727     <mets:name> National library of Finland </mets:name>
728 [...]
729 </mets:metsHdr>
730
731 SIP submitter:
732
733 <mets:metsHdr CREATEDATE="2018-02-11T08:00:00" RECORDSTATUS="NEW">
734   <mets:agent ROLE="PRESERVATION" TYPE="ORGANIZATION">
735     <mets:name> Kansalliskirjasto </mets:name>
736     <mets:note> ISNI 0000 0001 2033 7602 </mets:note>
737  [...]
738 </mets:metsHdr>
739
740
```

741 ### 6.2   Package status

742 The METS header RECORDSTATUS attribute value "REPLACEMENT" SHALL be used to indicate the
743 status of the package if the package is resubmitted. If the attribute is not present, its value is assumed to
744 be "NEW".

745 **Example**

746 Modified SIP to replace one sent earlier:

```
747    <mets:metsHdr CREATEDATE="2018-01-10T17:12:55" RECORDSTATUS="REPLACEMENT">
748    [...]
749    </mets:metsHdr>
750
```

6.3   Package identifier

Every SIP SHALL have a unique identifier. The submission agreement SHALL specify the identifier type or types used (for instance, UUID).

SIPs themselves are not preserved after the ingest process is finished, but the SIP identifier MAY be preserved both in the repository system and in producer's production systems, if there is a possibility the SIP identifier could be needed later on.

There are two encoding options, the first one of which is mandatory:

1) An identifier SHALL be located in the root element of the METS document using the OBJID attribute.
2) A SIP identifier MAY also be expressed in a PREMIS metadata record, if it is intended as a persistent identifier.

If a private identifier system is used, the name of the creator of the package (if the creator is not the producer) MAY be part of the identifier. This makes it possible to identify the creator, and the OAIS archive is able to contact that organization directly – instead of the producer – if there are technical problems during the ingestion process.

NOTE   METS uses the XML ID data type for identifiers. The requirement for this data type is that the first character of the value must be a letter. No numbers or other characters are allowed on the first position. Therefore it is reasonable to require a prefix starting with a letter (like the "urn:uuid:" below).

Examples

Package identifier in the root of a METS document:

```
772    <mets:mets OBJID="urn:uuid:A1B0D67E-2E81-4DF5-9E67-A64CBE366809"
773    xsi:schemaLocation="http://www.loc.gov/METS/ http://www.loc.gov/standards/mets/mets.xsd">
774    [...]
775    </mets:mets>
776
```

Publication identifier used as a package identifier in a Dublin Core record embedded in an EPUB publication:

```
779    <dc:identifier id="pub-id">urn:uuid:A1B0D67E-2E81-4DF5-9E67-A64CBE366809</dc:identifier>
780        <meta refines="#pub-id" property="identifier-type" scheme="xsd:string">uuid</meta>
781
```

Publication identifier SHALL NOT be used as package identifiers. A SIP can contain multiple EPUB publications; one EPUB publication can be submitted in multiple SIPs and even if a SIP contains just one publication it may be necessary to re-send the SIP with other package identifier.

Example

```
786    <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
787        <dc:identifier id="pub-id">urn:uuid:A1B0D67E-2E81-4DF5-9E67-
788    A64CBE366809</dc:identifier>
789        <meta property="dcterms:modified">2011-01-01T12:00:00Z</meta>
790        …
791    </metadata>
792
793    results in the Package ID:
```

794
795 `urn:uuid:A1B0D67E-2E81-4DF5-9E67-A64CBE366809@2011-01-01T12:00:00Z`
796

6.4   Publication identifiers

798 According to the EPUB specification, each EPUB publication SHALL have a unique identifier. However,
799 revised publications do not require a new standard identifier if only minor changes have been made,
800 such as metadata updates or errata fixes. In such cases, usage of release identifiers (which consist of e.g.
801 ISBN and the publication date) is recommended in the EPUB specification, but not mandatory.

802 In order to facilitate long-term preservation, each rendition and version of an EPUB publication
803 submitted to an OAIS archive SHALL have an identifier, and the submission agreement or other
804 guidelines SHALL specify the identifier systems allowed. If the archive's repository system cannot
805 process EPUB release identifiers (for instance because the system assumes each e-book has its own
806 ISBN or other standard identifier), release identifiers SHALL be replaced with standard identifiers
807 during pre-ingest by the producer.

808 Component parts of EPUB publications SHALL have separate identifiers if they are submitted as
809 independent publications. For instance, if each chapter of an e-book is submitted as a separate EPUB
810 publication, they SHALL have their own identifiers even if all chapters (EPUB publications) are sent in
811 the same SIP.

812 Producers MAY provide identifiers to fragments within publications, such as paragraphs or sentences
813 within a text. If such identifiers or other methods are used to provide links between for instance a text
814 and an audio version of the text within an EPUB 3 document, the OAIS archive SHALL maintain the links
815 even when the text or audio file is migrated into a new format, if submission agreement requires that
816 such functionality is retained and if the new audio format allows such linking.

817 Identifiers SHALL be persistent, globally unique and appropriate for the submitted content. For
818 example ISBN for books and DOI for scientific articles.

819 The submission agreement SHALL specify the encoding of publication identifiers. There are at least two
820 encoding options, of which one SHALL be selected in the agreement:

821   1)   An identifier is included in the Dublin Core metadata embedded in the EPUB publication, as
822   specified in the EPUB publications specification. The EPUB specification requires that the
823   identifier of a publication is provided in the Dublin Core element Identifier. The EPUB META
824   element MAY be used to indicate the identifier type, using an authorized list such as ONIX Code
825   List 5[13].
826   2)   Identifier is expressed using the <premis:objectIdentifier> element. If so, PREMIS encoding
827   SHALL specify the identifier type.

828 NOTE 1   Digital preservation systems may be unable to handle EPUB release identifiers. For
829   instance, their duplicate-check algorithms may expect ordinary use of standard identifiers
830   such as ISBNs. It is possible to build a digital preservation system optimized unique and
831   release identifiers for EPUB, but as of this writing no such systems exist in the library
832   sector.

833 NOTE 2   If the repository system cannot process release identifiers, it is possible to use a UUID or a
834   custom made system to identify individual renditions. The submission agreement should
835   specify the correct approach.

---

[13] http://www.stison.com/onix/codelists/onix-codelist-5.htm

836 NOTE 3   The submission agreement MAY also specify guidelines on the use of unique identifiers.
837   For instance, if ISBN is one of the supported identifier systems, the agreement MAY
838   specify that ISBNs SHOULD be used according to the ISBN manual.

839 **Examples**

840 Identifiers in a Dublin Core record (including an identifier for the resource itself and its source):

```
841 <dc:identifier id="isbn-id">urn:isbn:9780101010101</dc:identifier>
842     <meta refines="#isbn-id" property="identifier-type" scheme="onix:codelist5">15</meta>
843
844     <dc:source id="src-id">urn:isbn:9780375704024</dc:source>
845     <meta refines="#src-id" property="identifier-type" scheme="onix:codelist5">22</meta>
846
```

847 NOTE   There may be 1-n sources.

848 Identifier in PREMIS record:

```
849 <premis:objectIdentifier>
850   <premis:objectIdentifierType>urn</premis:objectIdentifierType>
851   <premis:objectIdentifierValue>
852     URN:ISBN:978-952-222-272-5
853   </premis:objectIdentifierValue>
854 </premis:objectIdentifier>
855
856 <mets:fileSec>
857   <mets:file ID="filee01" OWNERID="URN:ISBN:978-952-222-272-5" ...>
858     [...]
859   </mets:file>
860 </mets:fileSec>
861
```

862 Release identifier (unique identifier and the last modification date):

```
863 <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
864     <dc:identifier id="pub-id">urn:isbn:951-0-18434-9</dc:identifier>
865     <meta property="dcterms:modified">2016-05-03T12:00:00Z</meta>
866     …
867 </metadata>
868
869 Release identifier is 951-0-18434-9@2016-05-03T12:00:00Z
870
```

871 If an outdated EPUB publication is migrated during ingest to a more modern EPUB version or another e-
872 book format, the OAIS archive SHALL acquire a new identifier for the migrated publication. The
873 identifier type SHOULD not change; meaning that if the original e-book had an ISBN, the migrated one in
874 another format should receive an ISBN too (ISBN system requires that each manifestation of a book has
875 its own identifier). The archive MAY either request the new ISBN from the publisher, or assign its own
876 identifier, depending on the agreement made with the publisher / producer.

877 The metadata in an AIP SHALL contain both identifiers, even if the AIP only contained the migrated
878 document.

879 **6.5   Core media type resource identifiers**

880 Identifiers for core media type resources within an EPUB publication SHOULD be unique and persist as
881 long as the publication. They SHALL be included in the manifest file of the EPUB publication as specified
882 by the EPUB publications specification.

883 Best practice for digital preservation is to have all information (documents and metadata) in
884 standardized and widely used formats. If there is a core media type resource which has been specified
885 as non-archivable (for instance, a GIF image) in the submission agreement, it SHALL be migrated during

886 ingest, and the manifest file of the migrated publication SHALL be updated accordingly. The metadata in
887 the manifest within the EPUB in the AIP SHOULD contain identifiers for both the original and migrated
888 resource even if the AIP only contained the latter.

889 The EPUB remote resources property SHOULD NOT be allowed for core media type resources in
890 submission agreements because retrieval of these resources can fail during ingest. This would mean the
891 failure of the entire ingest process, because the archived EPUB publication would be incomplete.

892 NOTE   Modern repository systems allow ingest and storage of non-archivable file formats.

893 **Example**

894 Core media type resource identifiers in an EPUB manifest file:

```
895 <manifest>
896     <item id="nav"
897           href="nav.xhtml"
898           properties="nav"
899           media-type="application/xhtml+xml"/>
900     <item id="intro"
901           href="intro.xhtml"
902           media-type="application/xhtml+xml"/>
903     <item id="c1"
904           href="chap1.xhtml"
905           media-type="application/xhtml+xml"/>
906     <item id="c1-answerkey"
907           href="chap1-answerkey.xhtml"
908           media-type="application/xhtml+xml"/>
909     <item id="c2"
910           href="chap2.xhtml"
911           media-type="application/xhtml+xml"/>
912     <item id="c2-answerkey"
913           href="chap2-answerkey.xhtml"
914           media-type="application/xhtml+xml"/>
915     <item id="c3"
916           href="chap3.xhtml"
917           media-type="application/xhtml+xml"/>
918     <item id="c3-answerkey"
919           href="chap3-answerkey.xhtml"
920           media-type="application/xhtml+xml"/>
921     <item id="notes"
922           href="notes.xhtml"
923           media-type="application/xhtml+xml"/>
924     <item id="cover"
925           href="./images/cover.svg"
926           properties="cover-image"
927           media-type="image/svg+xml"/>
928     <item id="f1"
929           href="./images/fig1.jpg"
930           media-type="image/jpeg"/>
931     <item id="gif-f1"
932           href="./images/fig1.gif"
933           media-type="image/gif"/>
934     <item id="css"
935           href="./style/book.css"
936           media-type="text/css"/>
937     <item id="pls"
938           href="./speech/dict.pls"
939           media-type="application/pls+xml"/>
940 </manifest>
941
```

942 NOTE   These identifiers are only valid within a single EPUB publication and SHALL only be used
943   in that context. Therefore there is no requirement for uniqueness.

6.6 Foreign resource identifiers

Identifiers for embedded foreign resources SHOULD be unique within the publication. They SHALL be included in the manifest file of the EPUB publication as specified by the EPUB publications specification.

If there is a foreign resource which has been specified as non-archivable in the submission agreement, it SHALL be migrated during ingest, and the manifest file of the migrated publication SHALL be updated accordingly. The metadata in the manifest within the EPUB in the AIP SHOULD contain identifiers for both the original and migrated resource even if the AIP only contained the latter.

The EPUB remote resources property SHOULD NOT be allowed in submission agreements because retrieval of these resources can fail during ingest. This would mean the failure of the entire ingest process, because the archived EPUB publication would be incomplete.

Example

Foreign resource identifiers in an EPUB manifest file. Note that it is obligatory to specify the media type of these resources:

```
<manifest>
    <item id="item1"
          href="chap1_docbook.xml"
          media-type="application/docbook+xml"
          fallback="fall1"/>
    <item id="fall1"
          href="chap1.xml"
          media-type="application/z3986-auth+xml"
          fallback="fall2" />
    <item id="fall2"
          href="chap1.xhtml"
          media-type="application/xhtml+xml"/>
    …
</manifest>
```

NOTE 1   The fallback chain should terminate with a core media type (xhtml).

NOTE 2   The fallback mechanism is relevant for the preservation of EPUB resources in the long run. When an EPUB publication is preserved, the old and the new representation can be linked via the fallback chain. Those users who are still able to render the original publication can still use that, while others can use the latest version.

6.7 Identifiers for metadata records

A metadata record can be for instance a Dublin Core record or a PREMIS record embedded or linked to a SIP.

There SHALL be a unique and persistent identifier for each metadata record in a SIP. If possible, the identifier SHOULD be embedded in the identified record, using an appropriate metadata element (e.g. record identifier). This approach is not applicable for Dublin Core metadata records, since the format does not have a metadata identifier element. Private Dublin Core extensions that allow encoding of record identifiers SHOULD NOT be used.

There are at least three encoding options for metadata records, one of which SHALL be selected:

- Metadata records are embedded in a METS document within SIP using METS mdWrap elements.
- Metadata records are embedded in a SIP, with mdRef links from the METS file.
- Metadata records are external, linked to SIP using METS mdRef element.

990 If metadata is external, the repository system SHALL be able to retrieve the metadata records during
991 ingest.
992

993 NOTE   E-ARK specification requires embedded metadata in SIP, but not in METS file. This approach
994   was chosen since compared with METS option it is more flexible, easier for producers and
995   diminishes the risk of the METS file becoming too large to manage and use, especially if SIPs
996   may contain several publications.

997 The metadata wrapper element <mdWrap> provides a wrapper around metadata within a METS
998 document. Such metadata can be in one of two forms:
999

1000   1) XML encoded metadata, where the XML encoding is identified as belonging to a namespace other
1001   than the METS document namespace.
1002   2) Any arbitrary binary or textual form[14].

1003 The metadata reference element <mdRef> element is a generic element used throughout the METS
1004 schema to provide an indicator to metadata residing outside the METS document. The location of the
1005 metadata SHALL be recorded in the xlink:href attribute[15].

1006 Many metadata formats support metadata record identifiers such as LCCN (Library of Congress Card
1007 Number). If a metadata format is migrated during ingest, these identifiers SHALL be encoded so that
1008 there is no risk of mixing the publication identifiers and the metadata identifiers with one another.

1009 In a METS document this is easy since both the entire administrative metadata section (<amdSec>) and
1010 all its parts (technical metadata, <techMD>; intellectual property rights metadata, <rightsMD>; source
1011 metadata, <sourceMD>; and digital provenance metadata, digiprovMD>) can have identifiers of their
1012 own.

1013 If a PREMIS LINK is used to associate a metadata record with the rendition of an EPUB publication, the
1014 following syntax MAY be used:

```
1015 <package … prefix="premis: http://www.loc.gov/standards/premis/v3/index.html">
1016
1017     <metadata>
1018         …
1019         <link rel="textMD-record" href="http://example.org/textmd/12389347"/>
1020         …
1021     </metadata>
1022     …
1023 </package>
```

1024
1025 Metadata record identifiers SHALL be used whenever there is a possibility that either the producer or
1026 the OAIS archive updates descriptive or administrative metadata during the ingest process or long-term
1027 preservation.

1028 When an OAIS archive creates a new representation of an EPUB publication via migration (for instance,
1029 from EPUB 3.01 to EPUB 3.1), there are two representations of the same intellectual object, which
1030 means these representations (EPUB 3 publications) SHALL have different ISBNs[16]. However, the
1031 producer and the archive SHALL have an agreement on how to manage the migrations, since they have
1032 an impact on metadata as well. The technical metadata record will be new in every rendition, therefore
1033 there is no need to provide a link to the predecessor. Descriptive metadata will remain the same except
1034 for the changes made in the technical metadata elements. The access rights metadata should not change

---

[14] http://www.loc.gov/standards/mets/docs/mets.v1-9.html#mdWrap

[15] http://www.loc.gov/standards/mets/docs/mets.v1-9.html#mdRef

[16] The process of acquiring new standard identifiers, such as ISBN is usually specified in standards and user
guides. Archives have to follow the appropriate procedures when obtaining identifiers for migrated documents, or
request new identifiers from the producer.

1035 at all, since any changes in the copyright or licensing are likely to have the same impact on both
1036 representations. The preservation metadata record will be enriched with the migration event
1037 information and information about the agents (human and software) related to it. The updated
1038 metadata record applies only to the latest representation of the EPUB publication.

1039 The descriptive metadata record of a migrated document SHALL include the identifier of the original
1040 publication in an appropriate metadata element such as the Dublin Core element Source in order to
1041 enable linking between different manifestations of the resource, both in the repository system (in case
1042 these manifestations are in different AIPs), and in the producer's information systems. Producers can
1043 delete the original version of the document from the production systems and only keep the migrated
1044 version, because if needed the original can be retrieved from the repository system as a DIP, except if
1045 the repository system deletes the original version too. The best practice – that SHOULD be documented
1046 in the submission agreement - is to keep all of the versions of the resource in the OAIS archive if file size
1047 is not an issue to the capacity of the repository system.

1048 **Examples**

1049 Identifier for a preservation metadata record in a PREMIS format:

```
1050 <mets:digiprovMD ID="file2345AMDdprov01M">
1051     metsmdWrap MIMETYPE="text/xml" MDType="PREMIS" LABEL="PREMIS preservation metadata">
1052         <mets:xmlData>
1053             ….
1054         </mets:xmlData>
1055     <mets:mdWrap>
1056
```

1057 Identifier for a technical metadata record in a TextMD format embedded within a METS document:

```
1058 <mets:techMD ID="AMDTech01M">
1059     metsmdWrap MIMETYPE="text/xml" MDType="TextMD" LABEL="Technical Metadata for Text">
1060         <mets:xmlData>
1061             ….
1062         </mets:xmlData>
1063     <mets:mdWrap>
1064
```

1065 Link to an external ONIX metadata record from an EPUB publication:

```
1066 <link rel="onix-record" href="http://example.org/meta/records/onix/121099"/>
1067
```

1068 **6.8   Dates**

1069 There are many dates that may be relevant for EPUB publications in general. For instance, ONIX codelist
1070 issue 38 has 18 codes just for publishing dates (list 163, publishing date role)[17] including publication
1071 date, public announcement date, date of first publication, last reprint date and so on. From a digital
1072 preservation point of view, publishing date is important but there are also other important dates,
1073 including SIP creation and update dates, which SHALL be expressed in a machine understandable
1074 format and encoded in such a manner that there is no risk of confusion with other date information.

1075 In order to guarantee machine understandability, all dates and times SHALL be expressed using ISO
1076 8601[18]. The date or time given SHOULD be as accurate as possible and the time zone SHOULD be
1077 provided if needed (e.g. when the producer and the OAIS archive are on different time zones).

---

[17] http://www.editeur.org/files/ONIX%20for%20books%20-
%20code%20lists/ONIX_BookProduct_Codelists_Issue_38.html
[18] As of this writing the latest version of the standard is ISO 8601:2004. A new extended version of the standard is
currently under preparation. This document should always use the latest version of ISO 8601, although ISO/FDIS
8601 does contain a lot of new features, many of which have not been implemented yet.

21

6.8.1   Creation date of a Submission Information Package

1079
1080
The SIP creation date SHALL be present in the metadata. The date SHALL be provided in the package header (<mets:metsHdr>) using the CREATEDATE attribute.

Example

```
<mets:metsHdr CREATEDATE="2011-02-15T15:41:12">
[...]
</mets:metsHdr>
```

6.8.2   Modification date of a Submission Information Package

If there are severe problems in the SIP, such as missing mandatory metadata or unknown file formats, ingest will usually fail. When the revised SIP is re-submitted to the repository system, the last modification date SHALL be provided using the LASTMODDATE attribute alongside the original CREATEDATE attribute. In the initial package the RECORDSTATUS attribute is NEW, but the status of the resubmitted package SHOULD be MODIFIED. REPLACEMENT SHALL be used only when EPUB publication that has already been archived is replaced by a more modern edition. MODIFIED explains why LASTMODDATE is used.

The LASTMODDATE attribute MAY also be included if the SIP has not been submitted before, but the package has been under construction for a long time (at least several days).

Example

SIP creation date and modification date:

```
<mets:metsHdr CREATEDATE="2011-02-15T15:41:12" LASTMODDATE="2016-02-29T10:54:30+02:00">
[...]
</mets:metsHdr>
```

6.8.3   Creation/modification date of an EPUB publication

According to [Kasdorf],

> dc:date element SHALL be used to provide the date of the EPUB publication (not the publication date of a source publication, such as the print book from which the EPUB has been derived.

Publication date SHALL be provided in the ISO 8601 format:

```
YYYY-MM-DDThh:mm:ssZ
```

The precision of the date information varies; often just publication year is needed.

The last modification date of each rendition in an EPUB container is also a mandatory metadata element.

If there are two or more EPUB publications, or two or more renditions of an EPUB publication in one SIP, the dates SHALL be provided separately for each EPUB publication and rendition.

The last modification date is normally specified as the publication date. But if it is necessary to specify both the publication date and last modification date because the producer has modified the publication to meet the ingest requirements, the last modification date MAY be provided as a PREMIS dateCreatedByApplication element within the publication's PREMIS metadata record.

If one or more of the underlying core media files are migrated during ingest, the archived EPUB publication SHOULD get a new publication date.

1122 **Examples**

1123 EPUB publication date in Dublin Core:

```
1124 <dc:date>2016-01</dc:date>
```

1125 EPUB publication modification date in PREMIS:

```
1126 <premis:objectCharacteristics>
1127 [...]
1128   <premis:creatingApplication>
1129     <premis:dateCreatedByApplication>2016-02-15T15:43:03
1130     </premis:dateCreatedByApplication>
1131   </premis:creatingApplication>
1132 </premis:objectCharacteristics>
1133
```

1134 **6.8.4 Creation/modification of a metadata record**

1135 The date and time when a metadata record embedded in a SIP was created or last modified SHOULD be
1136 provided using a CREATED attribute in the appropriate METS metadata section (techMD etc.). The
1137 attribute requires precision down to a second; if the data is not accurate enough, the date can be
1138 padded with zeros if required. If even the specific date is unknown, the first of January (01-01) can be
1139 used instead.

1140 NOTE   This recommendation differs from the one stated in ISO/FDIS 8601 so reassessment may
1141   be necessary once the ISO 8601 revision has been concluded.

1142 The METS CREATED attribute SHOULD also be used when a metadata record has been modified.

1143 Some metadata formats allow expression of creation and modification dates in the metadata record
1144 itself. For instance, a MARC record contains the date the record was created in a fixed length field 008,
1145 positions 00-05, format YYMMDD. This information is never changed. Date and time of the last
1146 transaction (the time the record was last modified) is stored in the field 005, in a format
1147 yyyymmddhhmmss.ff, where the ff represents the decimal fractions of a second.

1148 Producers and OAIS archives MAY agree to use these metadata record creation and modification dates.

1149 **Example**

1150 Metadata record creation/modification date:

```
1151 <mets:dmdSec ID="dmd-dc" CREATED="2015-02-15T00:00:00">
1152 [...]
1153 </mets:dmdSec>
1154
```

1155 **6.9 Metadata format and its versions**

1156 This section is based on Bill Kasdorf's EPUB 3 Packaging and Metadata, which provides guidelines for
1157 supplying metadata in an EPUB 3 publication.

1158 EPUB 3 uses the Dublin Core Metadata Element Set for much of its required and optional metadata.
1159 There are three mandatory (Dublin Core) metadata elements (title, identifier, language), which SHALL
1160 be embedded in all EPUB 3 publications.

1161 The terms META, LINK, ITEM, and ITEMREF MAY be used to describe properties of key elements.
1162 Dublin Core metadata in an EPUB can be either simple or qualified; since the syntax for qualification is

1163   EPUB specific, it is possible that applications parsing the embedded Dublin Core records will "dumb"
1164   them down.

1165   **Example**

```
1166  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
1167  […]
1168  dc:identifier id="pub-identifier">urn:isbn:9781449325299</dc:identifier>
1169  <dc:title id="pub-title">EPUB 3 Best Practices</dc:title>
1170  <dc:title id="t2">First Edition</dc:title>
1171     <meta refines="#t2" property="title-type">edition</meta>
1172  <dc:language id="pub-language">en</dc:language>
1173  </metadata>
1174
```

1175   This metadata SHOULD be copied into a METS document, because some long-term preservation
1176   applications may not be able to retrieve this metadata from an EPUB publication.

1177   In a METS container, descriptive metadata (in the Dublin Core format) SHALL be expressed in a
1178   <mets:dmdSec> element using MDTYPE value "DC"[19].

1179   The version of the Dublin Core format MAY be expressed using an MDTYPEVERSION attribute.

1180   **Example**

1181   Original metadata record in an EPUB container:

```
1182  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
1183          <dc:identifier id="pub-id">urn:uuid:A1B0D67E-2E81-4DF5-9E67-
1184  A64CBE366809</dc:identifier>
1185          <dc:title>Norwegian Wood</dc:title>
1186          <dc:language>en</dc:language>
1187          <meta property="dcterms:modified">2011-01-01T12:00:00Z</meta>
1188      </metadata>
1189
```

1190   Simple Dublin Core metadata record embedded in a METS document. Note that the modification date of
1191   the record is included, encoded with an EPUB META term:

```
1192  <mets:dmdSec ID="dmd-dc" CREATED="2016-05-03T14:00:00">
1193    <mets:mdWrap MIMETYPE="text/xml"
1194         MDTYPE="DC"
1195         MDTYPEVERSION="1.1"
1196         LABEL="Bibliographic metadata">
1197      <mets:xmlData>
1198        <dc:record>
1199          <dc:identifier id="pub-id">urn:uuid:A1B0D67E-2E81-4DF5-9E67-
1200  A64CBE366809</dc:identifier>
1201          <dc:title>Norwegian Wood</dc:title>
1202          <dc:language>en</dc:language>
1203          <meta property="dcterms:modified">2011-01-01T12:00:00Z</meta>
1204        </dc:record>
1205      </mets:xmlData>
1206
```

1207   An EPUB publication can contain one or more links to external metadata records which describe the
1208   resource. The EPUB metadata link vocabulary[20] supports MARC 21, MODS and ONIX. If the linked
1209   metadata records are not just alternative representations of embedded metadata (that is, if there is no
1210   descriptive metadata in the publication or if the embedded metadata is abridged), linked metadata
1211   record(s) SHOULD be retrieved and embedded into the SIP as part of the pre-ingest process. Another

---

[19] MDTYPE list used by Harvard university is available at:
http://hul.harvard.edu/mets/doc/edu/harvard/hul/ois/mets/Mdtype.html

[20] http://www.idpf.org/epub/vocab/package/link/

1212 option is for the OAIS archive to retrieve the linked metadata during ingest to ensure the AIP is
1213 complete. The submission agreement SHALL specify if external metadata is allowed.

1214 The process described above applies to all kinds of metadata. If essential metadata, be it descriptive or
1215 administrative, is not a part of the SIP but just linked to it, this metadata SHALL be retrieved either
1216 during a pre-ingest by the producer or during ingest by the OAIS archive.

1217 7 Administrative metadata

1218 This specification is incomplete since it does not cover the administrative metadata elements needed to
1219 preserve EPUB core media type resources[21]. The technical metadata required for the preservation has
1220 to be media type specific, and covering the mandatory metadata elements needed for text, still images,
1221 and audio files in a single document is not feasible. The metadata elements listed here are media type
1222 neutral and are always needed in long-term preservation, no matter what the media type or file format
1223 is.

1224 The preservation metadata data dictionary PREMIS[22] is used for preservation metadata. The Library of
1225 Congress has published guidelines for using PREMIS with METS[23] with the intention to "suggest
1226 common practices for encoding METS documents with PREMIS metadata for exchange purposes"
1227 [Guidelines]. You can find an example of a METS document using the profile at
1228 http://www.loc.gov/standards/premis/louis-2-0.xml.

1229 OAIS archives follow their own practices when they create AIPs, but common guidelines SHOULD be
1230 applied for SIPs and DIPs in order to guarantee interoperability. It is the responsibility of both
1231 producers and OAIS archives to apply recommended practices whenever possible. If preservation
1232 metadata is included into the SIP, then for interoperability reasons it SHOULD follow common
1233 guidelines.

1234 Preservation metadata requirements depend a lot on the complexity of the EPUB publications to be
1235 archived. An EPUB publication containing just text is easier to deal with than one containing for
1236 instance MPEG audio linked into the text. Publications where the layout has no impact on semantics are
1237 much easier to preserve in the long term than fixed layout EPUB publications. In fact, if the original
1238 layout of an EPUB publication has a major impact on its meaning, it can be difficult or impossible to
1239 preserve the publication in the long term, and therefore the publication SHOULD be migrated into a
1240 "normal" EPUB publication during pre-ingest.

1241 Submission agreements covering EPUB publications SHALL list not only ingestible media types (file
1242 formats), but also the EPUB properties that MAY be used. These lists are not static; they SHALL be
1243 maintained in cooperation with the producer and the OAIS archive, since:

1244 • EPUB core media type list is updated frequently, and the impact of changes (new media types,
1245 deprecated old ones) has to be checked.
1246 • EPUB core media type list only covers file formats (such as image/jpeg), not their different
1247 versions. If a new version (like TIFF 7.0) is introduced, it is necessary to decide if it is ingestible
1248 or archivable and can therefore be submitted.
1249 • An EPUB core media type may become non-archivable, or vice versa – a core media type
1250 previously regarded as non-archivable may become "acceptable".
1251 • When new versions of the EPUB core specification and related documents are published, it is
1252 important to check the impact they have on long-term preservation.

---

[21] https://idpf.github.io/epub-cmt/v3/
[22] https://www.loc.gov/standards/premis/
[23] https://www.loc.gov/standards/premis/guidelines-premismets.pdf

1253 **7.1 Technical metadata**

1254 **7.1.1 File formats and their versions**

1255 EPUB version used SHALL be specified in the <package> element of the EPUB publication's content.opf
1256 file.

1257 NOTE ePubCheck does not provide accurate EPUB versioning beyond the major number.

1258 **Example**

1259 ```
<package xmlns="http://www.idpf.org/2007/opf" unique-identifier="ean" version="3.0.1">
1260 . . .
1261 <package>
```

1262

1263 File formats present in an EPUB container embedded in a SIP, including both core media types and
1264 foreign resources, SHALL be indicated using PREMIS <premis:formatName> element. From the PREMIS
1265 encoding point of view, it makes no difference whether the resource is a core media type or not.

1266 Versions of file formats SHALL be expressed in the <premis:formatVersion> element if they are known.
1267 Reference to a file format registry such as PRONOM[24] MAY be added if it is necessary to provide access
1268 to the full details of the file format.

1269 File formats (but not the version) MAY also be expressed in METS using the MIMETYPE attribute of the
1270 <mets:file> element. The attribute requires the use of IANA MIME[25] media types.

1271 The composition level SHALL be encoded using PREMIS compositionLevel element. Its value SHALL be
1272 1 if the EPUB publication is considered to be a container. However, if the EPUB publication is seen as a
1273 file only, composition level 0 MAY be used.

1274 NOTE Any application capable of rendering EPUB 3 publications SHOULD be able to deal with all
1275 core media type resources. Whether foreign resources, included with a fallback
1276 mechanism, will also be preserved beyond bit level, depends on what the producer and
1277 the OAIS archive have agreed on. The submission agreement SHOULD specify all
1278 ingestible and archivable file formats a publisher will submit. If other file formats are
1279 included in SIPs, they SHALL be encoded so that they are not validated during ingest
1280 (otherwise the ingest process will fail) and the OAIS archive's preservation responsibility
1281 is limited to bit level.

1282 If core media type resources and other resources are preserved via migration, migrating even a single
1283 file means the entire EPUB 3 publication SHALL be updated and a new AIP created, with updated
1284 descriptive and administrative metadata. If emulation is the chosen preservation method, the EPUB
1285 publication itself is not modified, but each time hardware or software environment changes,
1286 preservation metadata changes and a new AIP SHALL be formed. Such metadata (EPUB specific things
1287 that must be recorded to facilitate emulation) and its modifications are beyond the scope of this
1288 document.

1289 **Examples**

1290 The EPUB version:

1291 ```
<mets:amdSec>
1292   <mets:techMD ID="fileepub01-techmd" CREATED="2015-05-31T09:54:43">
1293     <mets:mdWrap MDTYPE="PREMIS:OBJECT">
```

---

[24] https://www.nationalarchives.gov.uk/PRONOM/Default.aspx

[25] https://www.iana.org/assignments/media-types/media-types.xhtml

```
1294            <mets:xmlData>
1295              <premis:object xsi:type="premis:file">
1296                <premis:objectIdentifier>
1297                  [...]
1298                </premis:objectIdentifier>
1299                <premis:objectCharacteristics>
1300                <premis:compositionLevel>1</premis:compositionLevel>
1301                <premis:format>
1302                  <premis:formatDesignation>
1303                    <premis:formatName>application/epub+zip</premis:formatName>
1304                    <premis:formatVersion>3.1</premis:formatVersion>
1305                  </premis:formatDesignation>
1306                </premis:format>
1307
1308                [...]
1309              </premis:objectCharacteristics>
1310            </premis:object>
1311          </mets:xmlData>
1312        </mets:mdWrap>
1313      </mets:techMD>
1314  </mets:amdSec>
1315
1316  <mets:fileGrp>
1317    <mets:file ID="fileepub01" ADMID="fileepub01-techmd">
1318      [...]
1319    </mets:file>
1320  </mets:fileGrp>
```

1322   An EPUB Core media type resource:

```
1323  <mets:amdSec>
1324    <mets:techMD ID="filee01-techmd" CREATED="2015-04-30T019:22:43">
1325      <mets:mdWrap MDTYPE="PREMIS:OBJECT">
1326        <mets:xmlData>
1327          <premis:object xsi:type="premis:file">
1328            <premis:objectIdentifier>
1329              [...]
1330            <premis:objectIdentifier>
1331            <premis:objectCharacteristics>
1332            <premis:compositionLevel>0</premis:compositionLevel>
1333            <premis:format>
1334              <premis:formatDesignation>
1335                <premis:formatName>image/png</premis:formatName>
1336                <premis:formatVersion>1.2</premis:formatVersion>
1337              </premis:formatDesignation>
1338            </premis:format>
1339            [...]
1340          </premis:objectCharacteristics>
1341        </premis:object>
1342      </mets:xmlData>
1343    </mets:mdWrap>
1344  </mets:techMD>
1345  </mets:amdSec>
1346
1347  <mets:fileGrp>
1348    <mets:file ID="filee01" ADMID="filee01-techmd">
1349      [...]
1350    </mets:file>
1351  </mets:fileGrp>
```

1353   An Embedded foreign resource:

```
1354  <mets:amdSec>
1355    <mets:techMD ID="filee01-techmd" CREATED="2015-06-30T015:12:00">
1356      <mets:mdWrap MDTYPE="PREMIS:OBJECT">
1357        <mets:xmlData>
1358          <premis:object xsi:type="premis:file">
1359            <premis:objectIdentifier>
```

```
1360                [...]
1361            <premis:objectIdentifier>
1362            <premis:objectCharacteristics>
1363            <premis:compositionLevel>0</premis:compositionLevel>
1364            <premis:format>
1365              <premis:formatDesignation>
1366                <premis:formatName>image/bmp</premis:formatName>
1367                <premis:formatVersion>1</premis:formatVersion>
1368              </premis:formatDesignation>
1369            </premis:format>
1370            [...]
1371            </premis:objectCharacteristics>
1372          </premis:object>
1373        </mets:xmlData>
1374      </mets:mdWrap>
1375    </mets:techMD>
1376 </mets:amdSec>
1377
1378 <mets:fileGrp>
1379    <mets:file ID="filee01" ADMID="filee02-techmd">
1380      [...]
1381    </mets:file>
1382 </mets:fileGrp>
```

1384 EPUB publications in SIPs SHOULD contain resources in file formats not suitable for preservation if and
1385 only if the same resource is also included in an acceptable file format using a fallback mechanism.

1386 If a file is migrated during pre-ingest to a format or a version suitable for preservation before
1387 submitting it to an archive, the migration SHOULD be documented in the SIP as a PREMIS EVENT. The
1388 original file MAY be included in the SIP with appropriate encoding (to bypass validation). If an archive
1389 migrates files during ingest, a PREMIS EVENT record is created. Migration related documentation
1390 SHALL be stored in AIPs regardless of who performed the migration, the producer, an archive, or a
1391 third-party.

1392 The event encoding SHOULD contain the following metadata:

1393   o   Event identifier
1394   o   Timestamp: <eventDateTime>2016-04-05</eventDateTime>
1395   o   Event type: <eventType>migration</eventType>
1396   o   Event outcome: <eventOutcome>success</eventOutcome>
1397   o   Link to the agent/agents: <premis:agentName>Word 2016</premis:agentName>
1398   o   Link to the PREMIS Object of the source file
1399   o   Including the role in the event: <linkingObjectRole>source</linkingObjectRole>
1400   o   Link to the PREMIS object of the output file
1401   o   Including the role in the event: <linkingObjectRole>outcome</linkingObjectRole>

1403 NOTE   PREMIS does not contain a standardized event vocabulary.

1404 There MAY be several source and output files. A 1:1 relationship between them is not required.

1405 Foreign files to be ignored during ingest SHALL be encoded using the METS <file> element with a USE
1406 attribute "no file-format-validation". For instance:

1407 `<file USE="no-file-format-validation" …>`
1408
1409 The archive MAY choose to validate foreign files although there is no intention to preserve them except
1410 at bit level. If so, USE attribute "no-file-format-migration" may be used.

1412 **Example**

1413 An EPUB 2 document encoded for bit level preservation:

```
1414  <mets:amdSec>
1415    <mets:techMD ID="fileepub201-techmd" CREATED="2016-05-03T09:54:43">
1416      <mets:mdWrap MDTYPE="PREMIS:OBJECT">
1417        <mets:xmlData>
1418          <premis:object xsi:type="premis:file">
1419            <premis:objectIdentifier>
1420              [...]
1421            </premis:objectIdentifier>
1422            <premis:objectCharacteristics>
1423            <premis:compositionLevel>1</premis:compositionLevel>
1424            <premis:format>
1425              <premis:formatDesignation>
1426                <premis:formatName>application/epub+zip</premis:formatName>
1427                <premis:formatVersion>2</premis:formatVersion>
1428              </premis:formatDesignation>
1429            </premis:format>
1430
1431            [...]
1432            </premis:objectCharacteristics>
1433          </premis:object>
1434        </mets:xmlData>
1435      </mets:mdWrap>
1436    </mets:techMD>
1437  </mets:amdSec>
1438
1439  <mets:fileGrp>
1440    <mets:file ID="fileepub201" ADMID="fileepub201-techmd" USE="no-file-format-validation">
1441      [...]
1442    </mets:file>
1443  </mets:fileGrp>
1444
```

7.1.2 Digital signatures and checksums

Archives can use digital signatures in various ways:

- For **submission** to an archive. A producer (publisher or a third party submitting the data) MAY sign an object, which enables the archive to guarantee that the submitting party is correct even if the transmission channel is not reliable.
- For **dissemination** from an archive. The archive MAY sign an object to assert that it truly is the source of the DIP.
- For **archival storage.** An archive MAY want to store signed objects so that it is possible for third-parties such as other archives or the data producer to confirm the origin and integrity of the data.

This document concentrates on the submission of objects to an archive, other uses are not discussed.

If there is no secure transmission channel, SIPs containing EPUB publications SHALL be digitally signed. A digital signature is not mandatory if there are other ways to make sure the SIP comes from the correct source.

Checksums SHOULD be calculated both for EPUB containers and their contents before the SIPs are sent to an archive in order to enable integrity checks. If the checksum is calculated to the container only, it is not possible to know which component has changed. If signatures are created immediately after the EPUB publication is created the producer can make sure that the content is not changed unintentionally before it is submitted to an archive.

Checksums SHOULD also be calculated for the core media resource files and for any foreign resources in the EPUB container.

1467 The checksum SHALL be calculated by using an algorithm specified in the submission agreement.
1468 Recommended options include sha-224, sha-256, sha-384 and sha-512. md5 and sha-1 SHOULD NOT be
1469 used because they are no longer safe.

1470 There are at least three different ways to embed checksums in metadata, one of which SHALL be
1471 selected in the submission agreement. Usage of METS File element is recommended, since then the
1472 METS file can be used to validate the integrity of the package.

1473 • METS element File (<file>) has attributes CHECKSUM and CHECKSUMTYPE. The values allowed
1474 for the latter are HAVAL, MD5, SHA-256, SHA-384, SHA-512, TIGER and WHIRLPOOL.
1475 Recommended options are SHA-256, SHA-384, and SHA-512.
1476 • A PREMIS element <premis:fixity> with attributes messageDigestAlgorithm and messageDigest.
1477 In PREMIS, running a fixity-check program on an object to detect unauthorized changes is called
1478 an EVENT.
1479 • signatures.xml file in an EPUB container allows the encoding of signatures for EPUB
1480 publications, their renditions as a whole, or just their component parts.
1481
1482 Syntax example for signatures.xml file can be found in the EPUB Container Format[26].

1483 **Example**

1484 Checksum in a METS FILE element:

```
1485        <mets:file ID="epi01m" CHECKSUMTYPE="SHA-256"
1486        CHECKSUM="a5d6ecbfc51f37b26778b24586dc15bfae8a0872275c39c2e19c63a5917650b5">
1487        </mets:file>
1488
```

1489 Checksum in a PREMIS metadata record:

```
1490  <mets:amdSec>
1491    <mets:techMD ID="fileepub01-techmd" CREATED="2011-05-31T00:00:00">
1492      <mets:mdWrap MDTYPE="PREMIS:OBJECT">
1493        <mets:xmlData>
1494          <premis:object xsi:type="premis:file">
1495            [...]
1496          <premis:objectCharacteristics>
1497            <premis:compositionLevel>2</premis:compositionLevel>
1498            <premis:fixity>
1499              <premis:messageDigestAlgorithm>
1500                SHA-256
1501              </premis:messageDigestAlgorithm>
1502              <premis:messageDigest>
1503                a5d6ecbfc51f37b26778b24586dc15bfae8a0872275c39c2e19c63a5917650b5
1504              </premis:messageDigest>
1505            </premis:fixity>
1506            [...]
1507          </premis:objectCharacteristics>
1508          </premis:object>
1509        </mets:xmlData>
1510      </mets:mdWrap>
1511    </mets:techMD>
1512  </mets:amdSec>
1513
```

1514 7.2 Rights metadata

1515 The copyright status of an EPUB publication SHOULD be expressed as rights metadata. If the embedded
1516 core media and foreign resources are copyrighted, their rights metadata SHOULD also be included if and
1517 when relevant. For instance, copyright owner for a foreign resource can be different than the copyright

---

[26] http://www.idpf.org/epub/301/spec/epub-ocf.html#sec-container-metainf-signatures.xml

1518 owner of the EPUB publication, even if access and use regulations were the same for all components of
1519 the publication.

1520 If a copyrighted publication (with its component parts) is licensed for use, the rights metadata SHOULD
1521 provide basic information about the license. Details about the terms of the license MAY be provided by
1522 e.g. providing a link to the copy of the license on the web.

1523 Any legal restrictions on the use of the document SHOULD be described in the embedded rights
1524 metadata.

1525 If a SIP contains several renditions of an EPUB publication with different rights information, each
1526 publication SHALL have its own rights metadata record attached to the rendition to which the metadata
1527 applies.

1528 There are at least three different methods for providing copyright status and license information. One of
1529 them SHALL be specified as mandatory in the submission agreement. The options are:

1530 • PREMIS <rights> element.

1531 • METS rightsMD element

1532 • META-INF/rights.xml file as specified in the EPUB Open Container Format

1533 **Exampl**e

1534 Rights metadata in PREMIS record.

```
1535  <premis>
1536    <rights>
1537      <rightsStatement>
1538        <rightsBasis>Copyright</rightsBasis>
1539        <copyrightInformation>
1540         <copyrightStatus>Under copyright<copyrightStatus>
1541         <copyrightJurisdiction>fi<copyrightJurisdiction>
1542         <copyrightNote>Copyright expires 2022<copyrightNote>
1543        </copyrightInformation>
1544      </rightsStatement>
1545    </rights>
1546  </premis>
1547
```

1548 This PREMIS record SHALL cover both rights related metadata and license metadata, if license covers
1549 preservation actions as well (see below).

1550 **7.2.1 Preservation related rights**

1551 Preservation related rights cover things done in the archive, from ingest to preservation to
1552 dissemination. These rights SHALL be based on the submission agreement, if copyright and licensing
1553 terms do not apply to actions done within an archive.

1554 Submission agreements MAY restrict preservation related actions the archive personnel is entitled to
1555 carry out. These restrictions are usually not described in individual SIPs, unless the publications in the
1556 SIP require special treatment. If there is preservation related rights metadata in a SIP, it SHOULD
1557 override the regulations in the submission agreement. For instance, if the archive staff is normally
1558 allowed to carry out migrations to the publications submitted by a certain producer, the producer
1559 SHOULD be able to prevent that with appropriate preservation metadata in a SIP. Submission
1560 agreements MAY specify that the archive is not allowed to migrate documents from a certain produce.
1561 Then migrations SHALL be done by the producer or a trusted third-party, and if they are no longer
1562 capable of the task, the OAIS archive SHALL be able to do it.

1563 For instance, a national library MAY outsource long term preservation of its legal deposit EPUB
1564 publications, but the library MAY still want to carry out critical preservation actions such as migrations
1565 itself. The responsibility of the OAIS archive would be limited to bit level preservation of this content.

1566 These preservation related restrictions SHALL be specified in the preservation plan. For instance, the
1567 plan may state that the OAIS archive is not allowed to migrate any EPUB publications submitted by a
1568 certain producer (for instance, the national library). If this information is not present in SIPs, the
1569 archive SHOULD add it to AIPs as preservation metadata during ingest. If the preservation plan is
1570 revised, old guidelines are deprecated and the OAIS archive SHALL update the rights metadata in the
1571 relevant AIPs.

1572 In information packages, restrictions for preservation actions SHALL be expressed using the PREMIS
1573 Rights metadata format and encoding in the METS <rightsMD> element. PREMIS rights metadata MAY
1574 also be included in EPUB publications.

1575 A   controlled   vocabulary   maintained   by   The   Library   of   Congress   available   at
1576 http://id.loc.gov/vocabulary/preservation/actionsGranted.html   SHALL   be   used   to   describe   the
1577 preservation actions (as of 2016-07-21, these are delete, disseminate, migrate, modify, replicate, and
1578 use).

1579 **Example**

1580 Premis rights metadata record included in a METS file:

```
1581  <amdSec ID="rights">
1582    <rightsMD ID="preservation-rights1">
1583      <mdWrap MIMETYPE="text/xml" MDTYPE="PREMIS" LABEL="PREMIS Rights Schema">
1584        <premis>
1585          <rights>
1586        <rightsStatement>
1587              <rightsBasis>Submission agreement</rightsBasis>
1588              <rightsGranted>
1589        <act>Disseminate</act>
1590        <restriction>Disallow</restriction>
1591        <termOfGrant>
1592              <startDate>2016-08-01</startDate>
1593              <endDate>open</endDate>
1594        <termOfGrant>
1595        <act>Modify</act>
1596        <restriction>Disallow</restriction>
1597        <act>Migrate</act>
1598        <restriction>Disallow</restriction>
1599              </rightsGranted>
1600            </rightsStatement>
1601          </rights>
1602        </premis>
1603      </mdWrap>
1604    </rightsMD>
1605  </amdSec>
```

1606

1607 **7.3   Structural metadata**

1608 In this chapter it is necessary to discuss:

1609   • internal structure of EPUB publication(s), and
1610   • structure of the SIP, which may contain 0-n EPUB publications

1611

1612 EPUB Open Container Format (OCF) SHALL be used to describe the structure of EPUB publication, as
1613 specified in http://www.idpf.org/epub/31/spec/epub-ocf.html.

1614 METS structMap SHALL be used to describe the structure of a SIP. If the SIP contains several EPUB
1615 publications, structMap SHALL specify them and the order in which they are to be presented.

1616 Structural metadata in OCF and METS structMAP is complementary; the former does not need to be
1617 aware of the SIP, and the latter does not need to describe the internal structure of the EPUB publication.
1618 But if the same structural metadata is provided in both formats, it SHOULD NOT be contradictory.

1619 The internal structure of an EPUB publication SHALL, according to the EPUB standard, be specified in
1620 an EPUB Navigation Document in both human and machine readable format[27]. This information does
1621 not need to be replicated in the METS document.

1622 An OAIS archive capable of ingesting EPUB 3 documents SHALL have an EPUB validator in order to
1623 make sure that the ingested publications are well formed. The archive SHOULD also have the reader
1624 application/applications a producer recommends for rendering the EPUB publications it has
1625 submitted, in order to be able to check when necessary that it is possible to render the ingested
1626 publications correctly. However, some OAIS archives MAY just ingest the EPUB publications and leave it
1627 to the users to find an appropriate EPUB reader or readers.

1628 **Example**

1629 A SIP containing 6 versions of the same EPUB publication, arranged hierarchically in different folders
1630 according to the nature of the versions.

1631 Example has been adapted from Rutgers university's METS structural map guidelines document[28]:

```
1632 <structMap TYPE="logical">
1633 <div ID="div1" LABEL="EPUB-SIP" ORDER="1" TYPE="folder">
1634 <fptr FILEID="FILE001" CONTENTIDS="ID1"/>
1635 <div ID="div1.1" LABEL="Folder A" ORDER="1" TYPE="folder">
1636 <fptr FILEID="FILE002" CONTENTIDS="IDH2"/>
1637 <fptr FILEID="FILE003" CONTENTIDS="ID3"/>
1638 <div ID="div1.1.1" LABEL="Folder A.1" ORDER="1" TYPE="folder">
1639 <fptr FILEID="FILE004" CONTENTIDS="ID4"/>
1640 </div>
1641 </div>
1642 <div ID="div1.2" LABEL="Folder B" ORDER="2" TYPE="folder">
1643 <fptr FILEID="FILE005" CONTENTIDS="ID5"/>
1644 </div>
1645 <div ID="div1.3" LABEL="Folder C" ORDER="3" TYPE="folder">
1646 <fptr FILEID="FILE006" CONTENTIDS="ID6"/>
1647 </div>
1648 </div>
1649 </structMap>
```

1650
1651 Within an EPUB container the HTML 5 nav element provides structural information:

```
1652 <nav epub:type="lot">
1653     <h2>List of tables, broken down into individual groups, one per major section of the
1654 publication content</h2>
1655     <ol>
1656         <li><span>Tables in Chapter 1</span>
1657             <ol>
1658                 <li><a href="chap1.xhtml#table-1.1">Table 1.1</a>
1659                 </li>
1660                 <li><a href="chap1.xhtml#table-1.2">Table 1.2</a></li>
1661             </ol>
1662         </li>
1663         <li><span>Tables in Chapter 2</span>
1664             <ol>
```

---

[27] http://www.idpf.org/epub/301/spec/epub-contentdocs.html#sec-xhtml-nav

[28] https://rucore.libraries.rutgers.edu/collab/ref/spc_sawg_r7_0_file_hierarchy.pdf

```
1665                    <li><a href="chap2.xhtml#table-2.1">Table 2.1</a>
1666                    </li>
1667                    <li><a href="chap2.xhtml#table-2.2">Table 2.2</a></li>
1668                    <li><a href="chap2.xhtml#table-2.3">Table 2.3</a></li>
1669                </ol>
1670           </li>
1671        …
1672        <li><span>Tables in Appendix</span>
1673              <ol>
1674                    <li><a href="appendix.xhtml#table-a.1">Table A.1</a>
1675                    </li>
1676                    <li><a href="appendix.xhtml#table-a.2">Table B.2</a></li>
1677              </ol>
1678        </li>
1679     </ol>
1680 </nav>
1681
```

1682 Each EPUB publication in a SIP SHALL contain the complete table of contents in the EPUB navigation
1683 document, covering all levels of the document hierarchy (see
1684 http://www.idpf.org/accessibility/guidelines/content/nav/toc.php). This information is important
1685 from an accessibility point of view, and although it is not as such relevant for preservation, it is required
1686 for the completeness of the SIP.

7.4 Preservation metadata

1688 Preservation metadata is a means of describing all relevant events that have taken place during the
1689 document lifecycle prior to, during, and after the ingest to an archive. For instance, if a producer has
1690 migrated the submitted EPUB 3 publication from EPUB 2 or some other file format, the preservation
1691 metadata provides information about software (and if necessary, hardware) used, and changes in the
1692 content and layout of the original document. Often good quality preservation metadata in PREMIS
1693 format is not required within the SIP but only created during ingest at the archives.

1694 However, if the submitter is able to provide preservation metadata about (migration) events occurring
1695 during pre-ingest or even earlier, it helps to demonstrate provenance and authenticity and could help
1696 archives, especially if some issues, such as migration problems, occur during ingest. If such producer-
1697 generated preservation metadata cannot be migrated to PREMIS, it MAY be included in the METS
1698 administrative metadata section in its native format.

1699 Some preservation metadata elements such as checksums may be generated very early in the life span
1700 of a publication, but archive-quality preservation metadata is usually produced during pre-ingest or
1701 ingest. At this point, it SHALL be expressed in a PREMIS format. If there is preservation metadata about
1702 past events that cannot be migrated to PREMIS, such metadata MAY be included in the METS
1703 administrative metadata section in its native format.

1704 If the original (un-archivable) version of the publication is also submitted to the OAIS archive,
1705 producer-generated preservation metadata in the SIP SHOULD specify the differences between the two
1706 renderings of the publication. Such metadata is useful when a customer is deciding which version of the
1707 publication would serve his needs better. If and when the publication is migrated during preservation,
1708 similar metadata about format migration SHOULD also be created and stored in the new AIP.

1709 Preservation metadata in a SIP MAY contain local PREMIS event types created by the producer. The
1710 archive SHALL pass all this metadata into the AIPs unchanged during the ingest process, except if there
1711 is a need to normalize vocabulary used.

1712 In order to simplify the ingest process, the producer or the submitting organization MAY check the
1713 validity of the EPUB publications before submitting them. If the publication has been validated, there
1714 SHOULD be a PREMIS validation event record in the SIP, documenting the outcome of the validation
1715 process and the validation tool used. Archival ingest workflow SHALL include validation as one of the

1716 steps. If a submitted publication has already been validated succesfully, the archive MAY choose to
1717 bypass the validation step.

1718 NOTE   OAIS archives tend to not trust producer-generated file validations. So even if a producer has
1719   validated EPUB publications it has submitted, and recorded appropriate metadata about these
1720   validation events in SIPs, the archive may decide to validate the submitted publications again.

1721 Submission agreements MAY require producers to carry out validation. Validation tools to be used, such
1722 as EpubCheck[29] or Ace by DAISY[30], SHOULD be specified in the agreement as well. An example of
1723 PREMIS metadata for a successful validation event of an EPUB publication is provided below.

1724 If validation fails, the publication SHOULD NOT be submitted before the problem is fixed, unless the
1725 submission agreement allows the submission of EPUB publications which have not passed validation.
1726 Such an allowance MAY apply to certain kinds of validation problems only. For instance, it is possible
1727 that the validation fails even though the problematic publication is rendered successfully with EPUB
1728 reading systems the document has been designed for. If the validation fails but the publication is
1729 nevertheless ingested as such, the negative validation result SHALL be included in the AIP preservation
1730 metadata. If the producer submits a corrected EPUB publication, there is no need to store the validation
1731 result of the original document.

1732 Core media file resources and foreign resources in EPUB publications itself SHOULD be validated
1733 independently to make sure that they actually are in formats claimed. Once the OAIS archive is certain
1734 that these resources are in file formats suitable for either ingest or preservation, these resources are
1735 treated accordingly (ingestible resources are migrated to archivable formats, resources in archival
1736 formats are transferred to AIPs). Common validation tools like JHOVE[31] SHOULD be used whenever
1737 possible (that is, when tools the OAIS archive has do support the file formats to be processed).

1738 If the validation fails, the SIP SHOULD NOT be submitted before the problem is fixed unless the
1739 submission agreement allows the producer to submit SIPs with non-archivable resources in them.
1740 Resources in these file formats SHALL be encoded in such a way that no attempt is made to validate or
1741 migrate them during ingest.

1742 The common validation tools do not cover all file formats and validation results can be less than perfect.
1743 Therefore it can be difficult to validate some resources beyond just rendering them.

1744 Validation results SHALL be expressed using the <mets:digiprovMD> element with PREMIS events
1745 (<premis:event>). A list of event types and examples of their use are provided in Annex 2. The event
1746 type list is not complete, but it covers typical events that can occur during the lifecycle of preserved
1747 documents.

1748 **Examples**

1749 Validation:

```
1750  <mets:digiprovMD ID="ev001" CREATED="2016-03-08T00:00:00">
1751    <mets:mdWrap MDTYPE="PREMIS:EVENT">
1752      <mets:xmlData>
1753        <premis:event>
1754          <premis:eventIdentifier>
1755            <premis:eventIdentifierType>local</premis:eventIdentifierType>
1756            <premis:eventIdentifierValue>
1757              epub3validation-001
1758              </premis:eventIdentifierValue>
1759          </premis:eventIdentifier>
```

29 https://github.com/IDPF/epubcheck/releases

30 https://github.com/daisy/ace

31 http://jhove.openpreservation.org/

```
1760              <premis:eventType>validation</premis:eventType>
1761              <premis:eventDateTime>2016-03-08T11:12:13</premis:eventDateTime>
1762              <premis:eventOutcomeInformation>
1763                <premis:eventOutcome>success</premis:eventOutcome>
1764                 <premis:eventOutcomeDetail>
1765                   format="EPUB"; version="3.0.1"; result="Well-formed and valid"
1766                 </premis:eventOutcomeDetail>
1767              </premis:eventOutcomeInformation>
1768              <premis:linkingAgentIdentifier>
1769                <premis:linkingAgentIdentifierType>
1770                  local
1771              </premis:agentIdentifierType>
1772              <premis:linkingAgentIdentifierValue>
1773                epubvalidator-4
1774              </premis:agentIdentifierValue>
1775            </premis:linkingAgentIdentifier>
1776          </premis:event>
1777        </mets:xmlData>
1778      </mets:mdWrap>
1779  </mets:digiprovMD>
1780
1781  <mets:digiprovMD ID="ag001" CREATED="2016-03-08T00:00:00">
1782    <mets:mdWrap MDTYPE="PREMIS:AGENT">
1783      <mets:xmlData>
1784        <premis:agent>
1785          <premis:agentIdentifier>
1786            <premis:agentIdentifierType>local</premis:agentIdentifierType>
1787            <premis:agentIdentifierValue>
1788              epubvalidator-4
1789            </premis:agentIdentifierValue>
1790          </premis:agentIdentifier>
1791          <premis:agentName>EpubCheck 4.0.0</premis:agentName>
1792          <premis:agentType>software</premis:agentType>
1793        </premis:agent>
1794      </mets:xmlData>
1795    </mets:mdWrap>
1796  </mets:digiprovMD>
1797
1798  Rendering an EPUB 3 file:
1799
1800  <mets:digiprovMD ID="ev002" CREATED="2016-03-10T00:00:00">
1801    <mets:mdWrap MDTYPE="PREMIS:EVENT">
1802      <mets:xmlData>
1803        <premis:event>
1804          <premis:eventIdentifier>
1805            <premis:eventIdentifierType>local</premis:eventIdentifierType>
1806            <premis:eventIdentifierValue>
1807              epub3rendering-001
1808            </premis:eventIdentifierValue>
1809          </premis:eventIdentifier>
1810          <premis:eventType>rendering</premis:eventType>
1811          <premis:eventDateTime>2016-03-10T14:12:05</premis:eventDateTime>
1812          <premis:eventOutcomeInformation>
1813            <premis:eventOutcome>success</premis:eventOutcome>
1814             <premis:eventOutcomeDetail>
1815               format="EPUB"; version="3.1"
1816             </premis:eventOutcomeDetail>
1817          </premis:eventOutcomeInformation>
1818          <premis:linkingAgentIdentifier>
1819            <premis:linkingAgentIdentifierType>
1820              local
1821          </premis:agentIdentifierType>
1822          <premis:linkingAgentIdentifierValue>
1823            epubrender-1
1824          </premis:agentIdentifierValue>
1825        </premis:linkingAgentIdentifier>
1826        </premis:event>
1827      </mets:xmlData>
1828    </mets:mdWrap>
1829  </mets:digiprovMD>
```

```
1830
1831   <mets:digiprovMD ID="ag001" epub3:CREATED="2016-03-10">
1832     <mets:mdWrap MDTYPE="PREMIS:AGENT">
1833       <mets:xmlData>
1834         <premis:agent>
1835           <premis:agentIdentifier>
1836             <premis:agentIdentifierType>local</premis:agentIdentifierType>
1837             <premis:agentIdentifierValue>
1838               epubrender-1
1839             </premis:agentIdentifierValue>
1840           </premis:agentIdentifier>
1841           <premis:agentName>EPUBReader 1.5.0.8</premis:agentName>
1842           <premis:agentType>software</premis:agentType>
1843         </premis:agent>
1844       </mets:xmlData>
1845     </mets:mdWrap>
1846   </mets:digiprovMD>
1847
```

8 Structure of Submission Information Packages

This document does not pose any requirements on the specific structure of the SIP.

A producer or other party submitting EPUB publications to an archive SHALL assemble a SIP containing 1-n EPUB publications or just descriptive or administrative metadata about them. Multiple EPUB publications in the same SIP are allowed if they are parts of the same work; for instance, chapters in a book.

A SIP MAY contain 0-n representations of the submitted EPUB publication in different formats such as PDF/A, and their associated metadata. Instead of a publication itself a SIP MAY contain just metadata about it.

Each representation of a publication in a SIP SHALL have its own metadata.

A SIP SHALL contain descriptive metadata in Dublin Core (as required by the EPUB 3 specification) and administrative metadata in PREMIS. Metadata in other formats MAY be included, as specified in the EPUB and METS standards, submission agreement, and other ingest related agreements between the producer and the OAIS archive (if any).

If SIPs are sent over a network and the transmission channel used is not secure, SIPs SHALL be signed electronically. Submission agreement SHALL specify secure means of transmission; they may include electronic (e.g. Secure File Transfer Protocol) and traditional channels (e.g. DHL).

The archive MAY use the same package structure when sending documents back to the producer or other consumer as dissemination information packages (DIPs). However, a DIP does not necessarily contain all the data and metadata present in a SIP. If the preserved EPUB 3 documents and metadata have been migrated during the preservation, the archive usually submits the latest version unless the customer demands an older version.

9 Content of Submission Information Packages

A SIP SHOULD contain a manifest file specifying the content of the package using METS structMap and fileSec.

The name and the location of the file depends on the submission agreement. If EPUB manifest.xml file is used, it SHALL be located in the META-INF directory at the root level of the EPUB container file system.

1875   If METS manifest file is used and it is located in the METS container, its name SHOULD be mets.xml[32]. If
1876   other name is used, it SHALL be specified in the submission agreement.

1877   The EPUB manifest file manifest.xml SHALL be compliant with the EPUB Open Container Format
1878   requirements.

1879   The manifest file mets.xml SHALL use METS schema to encapsulate information of the files in the SIP.
1880   Supported METS version(s) SHOULD be specified in the submission agreement. As of this writing the
1881   latest version of the standard is 1.11.

1882   The manifest file mets.xml SHALL be compliant with the EPUB 3 METS profile.

1883   The character encoding of the mets.xml file SHALL be UTF-8, and the file SHALL be located in the root
1884   directory of a SIP.

1885   The mets.xml SHALL contain in the structMap element structural metadata needed for locating the
1886   EPUB 3 publications and other documents such as PDF/A versions of the publication in the SIP. The SIP
1887   SHALL also contain descriptive and administrative metadata required for ingest and archival of the
1888   EPUB publications and other documents in the package. Depending on the submission agreement, the
1889   producer may have to copy mandatory metadata elements from the EPUB publications to the mets.xml
1890   file. Otherwise the archive harvests such metadata directly from the EPUB publications during ingest.

1891   A SIP MAY contain additional metadata not required by this specification. If so, such metadata elements
1892   SHALL be encoded in an appropriate manner if there is a risk of confusion. For instance, if there are
1893   additional MARC 21 identifier elements (for e.g. related publications or related versions of the
1894   submitted publication), it SHALL be possible to tell them apart from the identifier of the submitted
1895   EPUB publication. If the SIP contains several EPUB publications, or versions of the EPUB publication in
1896   other file formats, it SHALL be possible to attribute the metadata records in SIP to correct documents.

1897   The internal structure of EPUB 3 containers does not need to be expressed in the mets.xml file (if
1898   present). Repository systems supporting EPUB 3 as the ingest and preservation format SHOULD have
1899   the tools with which to render EPUB 3 documents and embedded core media resources. Although the
1900   OAIS archive may not be able to view every EPUB publication, it SHOULD have appropriate viewer
1901   applications with which to check whether a certain EPUB publication is well formed. Such rendering
1902   implies that the structural metadata in the spine of the EPUB publication is used.

1903   If a digital signature is needed to prove the identity of the submitting organization, there are two
1904   possible approaches, one of which SHOULD be selected in the submission agreement.

1905   An EPUB publication MAY contain signatures.xml file within the META-INF directory at the root level of
1906   the container file system. This file, if present, contains the digital signatures of the EPUB container and
1907   its contents. Its schema is specified in the EPUB Open Container Format.

1908   If the EPUB container does not have a signatures.xml file, before the SIP is sent the producer SHALL
1909   create a file signature.sig as described below, especially if the transmission channel is not secure. The
1910   signature.sig file SHALL contain all the required signatures, created with specific algorithms.

1911   The archive SHOULD be able to check the authenticity of the publication or publications in a SIP by a
1912   checksum. Checksums algorithms to be used SHALL be specified in the submission agreement. Such
1913   algorithms SHALL be commonly used message digest algorithms such as SHA-256.

1914   The producer SHOULD[33] calculate the checksum for every file and add it in the preservation metadata. If
1915   the SIP does not contain checksums, the OAIS archive SHALL calculate them and add them to the AIP.

---

[32] Two options are provided to guarantee backwards compatibility with existing systems.

1916 The integrity of the submitted content is checked by comparing the checksum calculated in the archive
1917 to the existing checksums. If the checksums do not match, the SIP's integrity cannot be verified and the
1918 ingest fails. The archive sends an error report to the producer or submitting organization, who SHALL
1919 check the problematic file and resubmit it.

1920

---

[33] This requirement ought to be SHALL. But in practice, producers often fail to provide checksums, and making them
mandatory might mean that producers would not send their publications to the archive anymore. This situation can be
improved by providing to the publishers tool sets which simplify the task of creating SIPs.

39

1921 **Annex A**

1922 (Informative)

1923

1924 **Digital Signature**

1925 The data integrity and nonrepudiation can be verified by a digital signature. With the signature the data
1926 sender can be verified even if the transmission channel has been unreliable.

1927 According to the EPUB documentation, creating digital signatures for the EPUB container and its
1928 contents is not mandatory, but if signatures have been created, they SHALL be placed into a
1929 signatures.xml file located in the META-INF directory.

1930 If the archive is located in-house, or if a secure channel has been used during the submission, a digital
1931 signature is not mandatory, but the SIP SHALL contain checksums so that it is possible to detect
1932 corruption of data during transfer and initial storage at the producer's systems. Checksums SHOULD be
1933 created right after documents are completed; otherwise it is not possible to detect if an EPUB
1934 publication has changed unintentionally during the initial storage.

1935 There are various ways in which digital signatures can be created. The submission agreement SHOULD
1936 specify the procedure. One possible approach is the following, adopted in the Finnish National Digital
1937 Library project:

1938 The submitting organization digitally signs the manifest file with a PKCS#7[34] (Public Key Cryptography
1939 Standard no. 7) signature. PKCS#7 is used to sign messages in the Public Key Infrastructure[35].

1940 The signature is formed as follows:

1941 • A Checksum for the manifest.xml file is calculated by using one of the algorithms specified in the
1942 submission agreement. Common options include md5, sha1, sha224, sha256, sha384 and
1943 sha512. Checksums for the EPUB 3 files have to already be contained in the METS document so
1944 as to avoid the need to calculate them as well while forming the signature for manifest.xml.
1945 • A Checksum is added to a text file by writing the following into it: checked file's (manifest.xml)
1946 path relative to the SIP's root, algorithm used and the checksum. These data elements are
1947 separated by semicolons.
1948 • A File is signed by PKCS#7 signature using a certificate.
1949 • A File is saved into the SIP package's root with the file name "signature.sig" in S/MIME format.
1950
1951 The archive verifies the checksum in the "signature.sig" file with the signer's public key and calculates
1952 the checksum using the same algorithm as the sender. If these checksums match, the integrity of the
1953 mets.xml file has been maintained during the transfer to the archive, and the archive can be certain that
1954 the file was signed by the correct producer.

1955 Below is an example of a "signature.sig" file created in this way. In the example the checksum is
1956 calculated by SHA-1 algorithm that is signed by a PKCS7 signature.

```
1957  MIME-Version: 1.0
1958  Content-Type: multipart/signed; protocol="application/x-pkcs7-signature";
1959  micalg=sha1; boundary="----57E5EFE5F87ADB48166F35F180BE72AC"
1960
1961  This is an S/MIME signed message
1962
1963  ------57E5EFE5F87ADB48166F35F180BE72AC
```

---

[34] https://en.wikipedia.org/wiki/PKCS

[35] https://en.wikipedia.org/wiki/Public_key_infrastructure

```
1964    Content-Type: text/plain
1965
1966    ./mets.xml:sha1:effdb5f96a28acd2eb19dcb15d8f43af762bd0ae
1967
1968    ------57E5EFE5F87ADB48166F35F180BE72AC
1969    Content-Type: application/x-pkcs7-signature; name="smime.p7s"
1970    Content-Transfer-Encoding: base64
1971    Content-Disposition: attachment; filename="smime.p7s"
1972    MIIFKwYJKoZIhvcNAQcCoIIFHDCCBRgCAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3
1973    ...
1974    UY/I9QDibkW5qAUY00rN7Q1F+uAGB+twEg6un1SVdg==
1975    ------57E5EFE5F87ADB48166F35F180BE72AC--
1976
```

1977 **Annex B**
1978 (Informative)
1979
1980 **Events**

1981 Long-term preservation requires actions made by human or software agents. To preserve the archived
1982 documents, they are ingested, refreshed, repackaged, replicated, and migrated in the archive, and
1983 disseminated to users. These actions are PREMIS events.

1984 Taken together, PREMIS events targeting an archived EPUB publication form a transaction log: a history
1985 of what has been done to the publication, and by what/whom. Such history can persist longer than the
1986 archived documents themselves. Therefore, digital archives SHALL keep track of everything that is done
1987 to their contents, and investigate all exceptions. An unauthorized copying of an archived publication is
1988 an example of an event that SHALL be scrutinized by the archive management. The likely result of such
1989 scrutiny is deletion of the extra copy.

1990 Some preservation actions have an impact on the look and feel and/or even the intellectual content of
1991 the preserved documents. If and when such changes do take place, they SHALL be documented properly
1992 as part of an event description. That way archive users can decide which version of the relevant
1993 document suits their information needs best. Migrated version is the easiest to use, though it may not
1994 be authentic enough for a critical user. Migrating e.g. fixed layout publications or interactive
1995 publications may cause serious loss of semantics or functionality in the publications.

1996 PREMIS is the de facto tool with which events are documented in long-term preservation. The standard
1997 itself does not contain a list of event types, but such lists have been created or are maintained by e.g.
1998 The Library of Congress and Archivematica:

1999 http://id.loc.gov/vocabulary/preservation/eventType.html
2000 https://wiki.archivematica.org/PREMIS_metadata:_events
2001
2002 Some libraries have created their own event lists. For instance, Bibliothèque nationale de France (BnF)
2003 uses packageDelivery (delivery of an information package) event described above, packageCreation
2004 (SIP ready), ingestCompletion (AIP ready) and disseminationCompletion (DIP ready) events.

2005 See also

2006 http://premis-event-service.readthedocs.org/en/latest/technical_overview.html

2007 for an explanation of PREMIS events and their use.

2008 The list below is provided as an example of how diverse PREMIS events can be. It contains events from
2009 Archivematica and Library of Congress lists and some other sources. The <premis:eventDetail> syntax
2010 is from the Archivematica list, but it is also possible (and perhaps preferable) to describe all
2011 applications as PREMIS agents; that way the eventDetail would just refer to the agent.

disseminationCompletion	Dissemination Information Package (DIP) ready
packageDelivery	Delivery of an information package
packageCreation	Creation of a new information package (SIP)
ingestCompletion	Archival Information Package (AIP) ready

capture	The process whereby a repository actively obtains an object.
compression	The process of coding data to save storage space or transmission time.

creation	The act of creating a new EPUB publication. If this event is included, event outcome (usually success) SHALL be provided, and the name and version of the tool used SHOULD be specified in the PREMIS eventDetail element. For example:
	
	`<premis:eventDetail>`
	`program="AdobeInDesign";version="CC`
	`(2015)"`
	`</premis:eventDetail>`

deaccession	The process of removing an object from the inventory of a repository.
decompression	The process of reversing the effects of compression.
decryption	The process of converting encrypted data to plain text.
deletion	The process of removing an object from repository storage.
digital signature validation	The process of determining that a decrypted digital signature matches an expected value.
fixity check	The process of verifying that an object has not been changed in a given period.
ingestion	The process of adding objects to a preservation repository.
message digest calculation	The process by which a message digest ("hash") is created.

migration	A transformation of an object creating a version in a more contemporary or more widely used file format (for instance, migrating a DITA file into EPUB 3). If this event is included, the outcome SHALL be provided in eventOutcome, and the name and the version (when applicable) of the tool used SHOULD be added into PREMIS eventDetail element. For example:
	
	`<premis:eventOutcomeInformation>success`
	`</premis:eventOutcomeInformation>`
	`<premis:eventDetail>`
	`program="DITA for Publishers EPUB plugin"`
	`version="1.0.0RC16"`
	`</premis:eventDetail>`
	`<premis:eventOutcomeDetail>`
	`format="EPUB";version="3.0.1";`
	`result="success"`

	</premis:eventOutcomeDetail>

processing	The process of doing something for the object, to be used when there is no dedicated event type for an operation. A description of the process SHALL be added into the PREMIS eventDetail element. For example:
	
	`<premis:eventDetail>`
	`Deskew: 0.49 CW;`
	`Crop: left 24px, top 9px, right 29px, bottom 12px;`
	`</premis:eventDetail>`

rendering	The process of rendering the document; for instance, reading an EPUB 3 document.
replication	The process of creating a copy of an object that is, bit-wise, identical to the original.

validation	The process of validating an EPUB file. If this event is included, the name and the version (when applicable) of the tool used SHOULD be added into the PREMIS eventDetail element and the result of the process SHALL be provided as an eventOutcomeDetail element. For example:
	
	`<premis:eventDetail>`
	`program="EpubCheck"version="4.0.2"`
	`</premis:eventDetail>`
	`<premis:eventOutcomeInformation>pass`
	`</premis:eventOutcomeInformation>`
	`<premis:eventOutcomeDetail>`
	`format="EPUB";version="3.0.1";`
	`result="Well-formed and valid"`
	`</premis:eventOutcomeDetail>`

virus check	The process of scanning a file for malicious programs.

2012

2013   Usually the event outcome is a success (pass), failure, or unknown.

2014   Below is another example from the BnF, describing how delivery of an information package can be
2015   expressed as a PREMIS event.

2016   The producer is referred to with digiprovMD (ID="AMD.10"), which makes it possible to describe the
2017   producer as a PREMIS agent below. Note that in the BnF implementation, producers also have UUIDs.

```
2018   <digiprovMD ID="AMD.9"  ADMID="AMD.10">
2019     <mdWrap MIMETYPE="text/xml" MDTYPE="PREMIS:EVENT">
2020       <xmlData>
2021         <premis:event>
2022           <premis:eventIdentifier>
2023             <premis:eventIdentifierType>UUID</premis:eventIdentifierType>
2024             <premis:eventIdentifierValue>f082af00-85b4-11e2-98c4-00144f80ca6b
2025   </premis:eventIdentifierValue>
2026           </premis:eventIdentifier>
2027           <premis:eventType>packageDelivery</premis:eventType>
2028           <premis:eventDateTime>2016-01-25T11:22:16+01:00</premis:eventDateTime>
2029           <premis:eventDetail>Channel 215: initial delivery</premis:eventDetail>
2030
2031           <premis:linkingAgentIdentifier>
2032             <premis:linkingAgentIdentifierType>UUID</premis:linkingAgentIdentifierType>
```

```
2033            <premis:linkingAgentIdentifierValue>2ba05b98-a4c3-11e5-bf7f-feff819cdc9f
2034   </premis:linkingAgentIdentifierValue>
2035              <premis:linkingAgentRole>issuer</premis:linkingAgentRole>
2036            </premis:linkingAgentIdentifier>
2037
2038            <premis:linkingObjectIdentifier>
2039              <premis:linkingObjectIdentifierType>Order form
2040   </premis:linkingObjectIdentifierType>
2041              <premis:linkingObjectIdentifierValue>16-SI-0001
2042   </premis:linkingObjectIdentifierValue>
2043              <premis:linkingObjectRole>request</premis:linkingObjectRole>
2044            </premis:linkingObjectIdentifier>
2045
2046            <!-- fileGroup uses that are delivered  -->
2047            <premis:linkingObjectIdentifier>
2048              <premis:linkingObjectIdentifierType>USE</premis:linkingObjectIdentifierType>
2049              <premis:linkingObjectIdentifierValue>master</premis:linkingObjectIdentifierValue
2050   >
2051              <premis:linkingObjectRole>outcome</premis:linkingObjectRole>
2052            </premis:linkingObjectIdentifier>
2053            <premis:linkingObjectIdentifier>
2054              <premis:linkingObjectIdentifierType>USE</premis:linkingObjectIdentifierType>
2055              <premis:linkingObjectIdentifierValue>colorProfile</premis:linkingObjectIdentifie
2056   rValue>
2057              <premis:linkingObjectRole>outcome</premis:linkingObjectRole>
2058            </premis:linkingObjectIdentifier>
2059          </premis:event>
2060        </xmlData>
2061      </mdWrap>
2062   </digiprovMD>
2063   <digiprovMD ID="AMD.10">
2064      <mdWrap MIMETYPE="text/xml" MDTYPE="PREMIS:AGENT">
2065        <xmlData>
2066          <premis:agent>
2067            <premis:agentIdentifier>
2068              <premis:agentIdentifierType>UUID</premis:agentIdentifierType>
2069              <premis:agentIdentifierValue>2ba05b98-a4c3-11e5-bf7f-
2070   feff819cdc9f</premis:agentIdentifierValue>
2071            </premis:agentIdentifier>
2072            <premis:agentName>Producer name</premis:agentName>
2073            <premis:agentType>organization</premis:agentType>
2074          </premis:agent>
2075        </xmlData>
2076      </mdWrap>
2077   </digiprovMD>
2078
2079
```

Bibliography

[1]   CCSDS 650.0-M-2 Reference model for an Open Archival Information System (OAIS). [online]. [viewed 2016-05-03]. Available from http://public.ccsds.org/publications/archive/650x0m2.pdf .

[2]   Digital Preservation Handbook. 2nd, revised ed. [online]. Digital Preservation Coalition, 2017. Available from: http://www.dpconline.org/handbook. [viewed 2017-07-25].

[3]   E-ARK SIP Pilot Specification. Version D3.3 (revision of D3.2). [online]. [s.l.], Project E-Ark, 2016. Available from: http://www.eark-project.com/resources/project-deliverables/51-d33pilotspec. [viewed 2017-07-27].

[4]   EPUB Metadata Link Vocabulary. 26 April 2016. [online]. [viewed 2016-05-03]. Available from: http://www.idpf.org/epub/vocab/package/link/.

[5]   EPUB Open Container Format (OCF) 3.1. W3C Member submission 25 January 2017. [online]. Available from: https://www.w3.org/Submission/2017/SUBM-epub-ocf-20170125/. [viewed 2017-07-25].

[6]   EPUB 3.1. W3C Member submission 25 January 2017. [online] . Available from: https://www.w3.org/Submission/2017/SUBM-epub31-20170125/. [viewed 2017-07-25].

[7]   Guidelines for using PREMIS with METS for exchange. Revised September 17, 2008 [online].

[8]   Kasdorf, Bill: EPUB 3 packaging and metadata. [online]. [s.l.]: EPUBZone, 2013. Available from: http://epubzone.org/news/epub-3-packaging-and-metadata [viewed 2017-07-26]. Archived version available at: https://web.archive.org/web/20140701083126/http://epubzone.org/news/epub-3-packaging-and-metadata

[9]   Lavoie, Brian. Meeting the challenges of digital preservation: the OAIS reference model. [online]. Columbus, OH: OCLC, 2000. Available from: http://www.oclc.org/research/publications/library/2000/lavoie-oais.html. [viewed 2016-07-15].

[10]   Metadata requirements and preparing content for digital preservation. [online]. The National Digital Library Project, 2017. Available from: http://www.kdk.fi/images/tiedostot/NDL-Metadata_v1.6.1.pdf. [viewed 2017-07-25].

[11]   Understanding metadata. [online]. Bethesda, MD.: NISO, 2004. Available from: http://www.niso.org/publications/press/UnderstandingMetadata.pdf. [viewed 2016-07-20].