

Executive Summary

This base profile describes the Common Specification for Information Packages (CSIP) and the implementation of METS for packaging OAIS conformant Information Packages. The profile is accompanied with a textuall document explaining the details of use of this profile. This will enable repository interoperability and assist in the management of the preservation of digital content. This profile is a base profile which is extended with E-ARK implementation of SIP, AIP and DIP. The profile can be used as is, but it is recommended that the supplied extending implementation are used. Alternatively, an own extension fulfilling the extending needs of the implementer can be created.

Table of contents

Preface	5
I. Aim of the specification	5
Overview of the E-ARK Specifications	6
II Organisational support	8
III Authors & Revision History	8
E-ARK CSIP Table Of Contents	8
Contents	8
1 Introduction	9
1.1 The Common Specification for Information Packages and OAIS	10
1.2 The Common Specification for Information Packages and Content Information Type Specifications	12
1.3 Common Specification for Information Packages, OAIS Information Packages' specifications and Content Information Type Specifications	13
1.4. Relation to other documents	14
International standards and best-practices	14
Other E-ARK specifications	14
1.5. Structure of the document	15
PART I: Common Specification for Information Packages	16
2 Need for establishing common ground	16
3 Principles for interoperable Information Packages	19
3.1 General principles	19
Principle 1.1	19
Principle 1.2:	20
Principle 1.3	20
Principle 1.4:	20
Principle 1.5:	21
Principle 1.6:	21
Principle 1.7:	21
3.2 Identification of the Information Package	22
Principle 2.1:	22
Principle 2.2:	22
Principle 2.3:	22

Principle 2.4:	22
Principle 2.5:	23
3.3 Structure of the Information Package	23
Principle 3.1:	23
Principle 3.2:	24
Principle 3.3:	24
Principle 3.4:	24
Principle 3.5:	24
Principle 3.6:	25
3.4 Information Package Metadata	26
Principle 4.1:	26
Principle 4.2:	26
Principle 4.3:	27
PART II: Implementation of the CSIP	27
4. CSIP structure	28
4.1. Folder structure of the CSIP	28
4.2. Implementing the structure	30
5. Use of metadata	32
5.1. General requirements for metadata in a CSIP Information Package	32
5.2 General requirements for the use of metadata	33
The use of identifiers	33
Referencing between files within a CSIP Information Package	34
Referencing other packages	34
Examples	35
5.3 Use of METS	35
5.3.1. Use of the METS root element (element mets)	36
5.3.2. Use of the METS header (element metsHdr)	39
5.3.3 Use of the METS descriptive metadata section (element dmdSec)	41
5.3.4. Use of the METS administrative metadata section (element amdSec)	43
5.3.5. Use of the METS file section (element fileSec)	49
5.3.6. Use of the METS structural map (<structMap>)	57
5.4. Use of PREMIS	67
Vocabularies	69
Identifiers	69
6. Implementation Considerations	69

6.1 Content Information Type Specifications	69
6.1.1 What is a Content Information Type Specification?	69
6.1.2 Maintaining Content Information Type Specifications	70
6.2. Handling large packages	71
6.3 Handling descriptive metadata within the Common Specification	71
Appendices	72
Appendix A: E-ARK Information Package METS examples	72
Appendix B: External Schema and Vocabularies	80
External Schema	80
E-ARK CSIP METS Extension	80
PREMIS	80
Controlled Vocabularies	80
Content information type specification	80
Content Category	80
OAIS Package type	80
Note type	81
Other agent type	81
Identifier type	81
dmdSec status	81
IANA media types	81
File group names	81
Structural map typing	82
Structural map label	82
Appendix C: A Full List of E-ARK CSIP Requirements	82
Postface	103
I Authors	103
Contributors to previous version	104
II Revision History	104

Preface

I. Aim of the specification

This specification is one of several related specifications. The single most important aim of all of these specifications is the provision of a common set of specifications for packaging digital information for archiving purposes. The specifications are based on common, international standards for

transmitting, describing and preserving digital data. They have been produced to help data creators, software developers and digital archives to tackle the challenge of short-, medium- and long-term data management and reuse in a sustainable, authentic, cost-efficient, manageable and interoperable way.

The foundation upon which the specifications are built is the Reference model for an Open Archival Information System (OAIS) (OAIS Reference model) which has Information Packages as its basis. Familiarity with the core functional entities of OAIS is a prerequisite for understanding the specifications. A visualisation of the current specification network can be seen here:

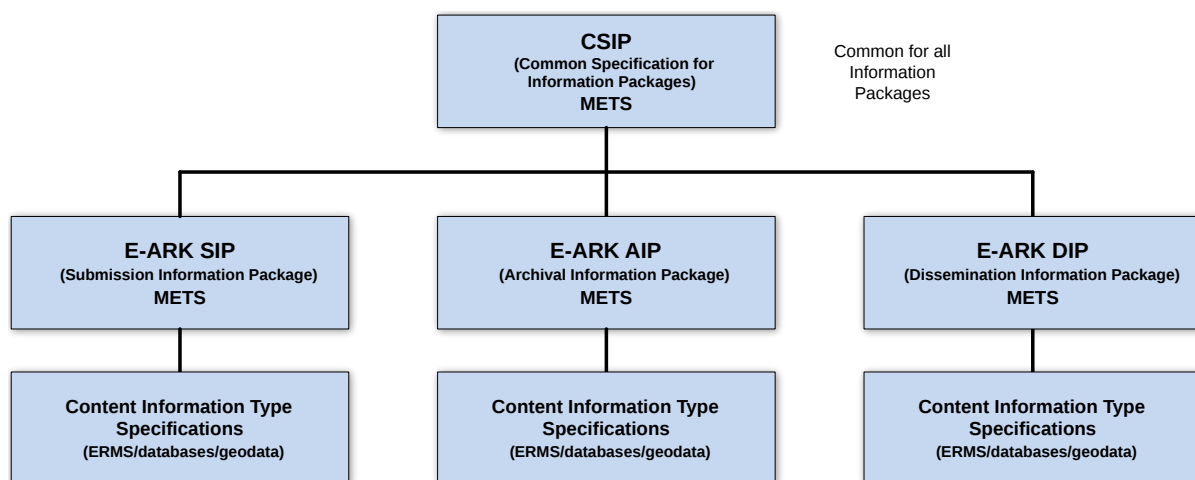


Figure I: Diagram showing E-ARK specification dependency hierarchy.

Overview of the E-ARK Specifications

Common Specification for Information Packages (E-ARK CSIP)

This document introduces the concept of a Common Specification for Information Packages (CSIP). Its three main purposes are to:

- Establish a common understanding of the requirements which need to be met in order to achieve interoperability of Information Packages.
- Establish a common base for the development of more specific Information Package definitions and tools within the digital preservation community.
- Propose the details of an XML-based implementation of the requirements using, to the largest possible extent, standards which are widely used in international digital preservation.
- Ultimately the goal of the Common Specification is to reach a level of interoperability between all Information Packages so that tools implementing the Common Specification can be adopted by institutions without the need for further modifications or adaptations.

Specification for Submission Information Packages (E-ARK SIP)

The main aims of this specification are to:

- Define a general structure for a Submission Information Package format suitable for a wide variety of archival scenarios, e.g. document and image collections, databases or geographical data.
- Enhance interoperability between Producers and Archives.
- Recommend best practices regarding metadata, content and structure of Submission Information Packages.

Specification for Archival Information Packages (E-ARK AIP)

The main aims of this specification are to:

- Define a generic structure of the AIP format suitable for a wide variety of data types, such as document and image collections, archival records, databases or geographical data.
- Recommend a set of metadata related to the structural and the preservation aspects of the AIP as implemented by the reference implementation (earkweb).
- Ensure the format is suitable to store large quantities of data.

Specification for Dissemination Information Packages (E-ARK DIP)

The main aims of this specification are to:

- Define a generic structure of the DIP format suitable for a wide variety of archival records, such as document and image collections, databases or geographical data.
- Recommend a set of metadata related to the structural and access aspects of the DIP.

Content Information Type Specifications (E-ARK CITS)

The main aims of a Content Information Type Specification are to:

- Define, in technical terms, how data and metadata must be formatted and placed within a CSIP Information Package in order to achieve interoperability in exchanging specific Content Information.
- The number of possible Content Information Type Specifications is unlimited. For a list of existing Content Information Type Specifications see, and read more about Content Information Type Specifications in the Common Specification for Information Packages.

II Organisational support

This specification is maintained by the Digital Information LifeCycle Interoperability Standards Board (DILCIS Board, <http://dilcis.eu/>). The DILCIS Board was created to enhance and maintain the draft specifications developed in the European Archival Records and Knowledge Preservation Project (E-ARK project, <http://eark-project.com/>) which concluded in January 2017. The Board consists of eight members, but there is no restriction on the number of participants in the work. All Board documents and specifications are stored in GitHub (<https://github.com/DILCISBoard/>) while published versions are made available on the Board webpage. Since 2018 the DILCIS Board has been responsible for the core specifications in the Connecting Europe Facility eArchiving Building Block <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eArchiving/>.

III Authors & Revision History

A full list of contributors to this specification, as well as the revision history can be found in the Postface material.

E-ARK CSIP Table Of Contents

Version: 2.0.0

May 31, 2019

Contents

- 1 Introduction
 - 1.1 The Common Specification for Information Packages and OAIS
 - 1.2 The Common Specification for Information Packages and Content Information Type Specifications
 - 1.3 Common Specification for Information Packages, OAIS Information Packages' specifications and Content Information Type Specifications
 - 1.4. Relation to other documents
 - * International standards and best-practices
 - * Other E-ARK specifications
 - 1.5. Structure of the document
- PART I: Common Specification for Information Packages
 - 2. Need for establishing common ground

- 3. Requirements for CSIP Information Packages
 - * 3.1. General principles
 - * 3.2. Identification of the Information Package
 - * 3.3. Structure of the Information Package
 - * 3.4. Information Package Metadata
- PART II: Implementation of the CSIP
 - 4. CSIP structure
 - * 4.1 Folder structure of the CSIP
 - * 4.2 Implementing the structure
 - 5. Use of metadata
 - * 5.1. General requirements for metadata in a CSIP Information Package
 - * 5.2. General requirements for the use of metadata
 - * 5.3. Use of METS
 - * 5.3.1. Use of the METS root element (element mets)
 - * 5.3.2. Use of the METS header (element metsHdr)
 - * 5.3.3 Use of the METS descriptive metadata section (element dmdSec)
 - * 5.3.4. Use of the METS administrative metadata section (element amdSec)
 - * 5.3.5. Use of the METS file section (element fileSec)
 - * 5.3.6. Use of the METS structural map (element structMap)
 - * 5.4. Use of PREMIS
 - 6. Implementation considerations
 - * 6.1. Content Information Type Specifications
 - 6.1.1. What is a Content Information Type Specification?
 - 6.1.2. Maintaining Content Information Type Specifications
 - * 6.2. Handling large packages
 - * 6.3. Handling descriptive metadata within the Common Specification
- Appendices
 - Appendix A: E-ARK Information Package METS examples
 - Appendix B: External Schema and Vocabularies
 - Appendix C: A Full List of E-ARK CSIP Requirements
- Postface

1 Introduction

This document introduces the concept of the Common Specification for Information Packages (CSIP). It aims to serve three main purposes:

1. Establish a common understanding of the requirements which need to be met in order to achieve

interoperability of Information Packages;

2. Establish a common base for the development of more specific Information Package definitions and tools within the digital preservation community;
3. Propose the details of an XML-based implementation of the requirements using, to the largest possible extent, standards which are widely used in international digital preservation.

Ultimately the goal of the Common Specification for Information Packages is to provide a level of interoperability between Information Packages so that tools implementing the CSIP can be taken up by institutions without the need for major modifications or adaptations.

1.1 The Common Specification for Information Packages and OAIS

The OAIS framework defines three types of Information Packages (IPs) present in the digital preservation ecosystem: Submission Information Packages (SIPs), Archival Information Packages (AIPs) and Dissemination Information Packages (DIPs) (Figure 1). These three IP types are respectively used to submit data and metadata to digital repositories; store it in long-term preservation facilities; and deliver to consumers.

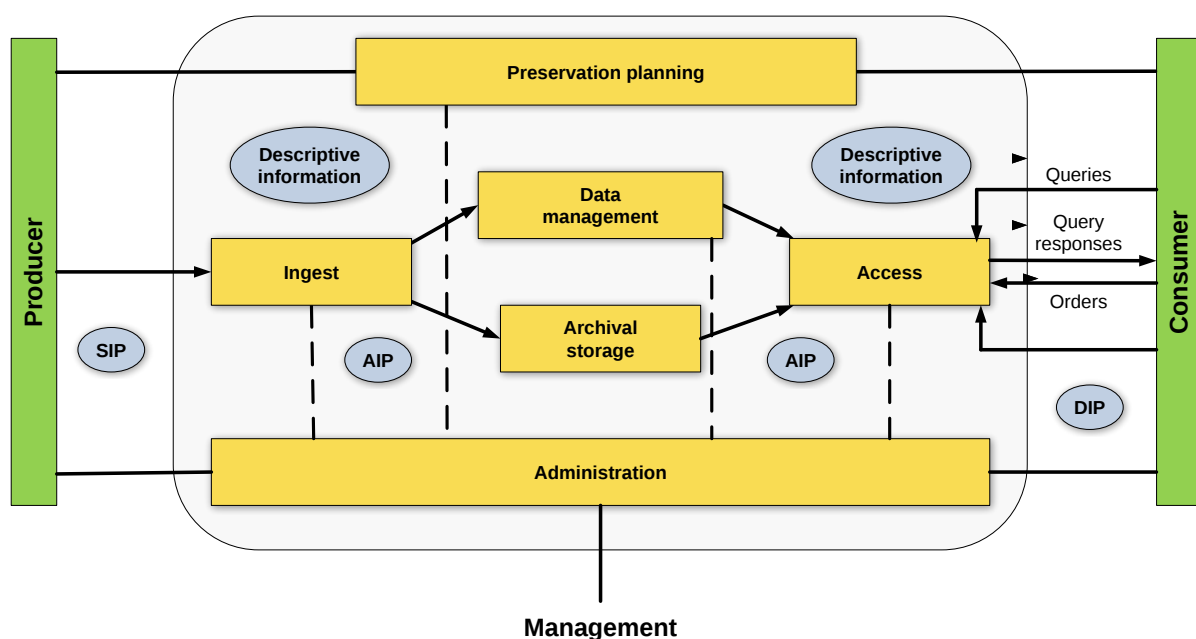


Figure 1: OAIS Functional Entities and Information Packages

The main goal in the development of this specification has been to identify and standardise the common aspects of IPs which are equally relevant and implemented by any of the functional entities of the overall digital preservation process presented in OAIS (i.e. pre-ingest, ingest, archival storage, data management and access). The practical implementation is that the specification allows for

the development of generic tools and code libraries which can either be applied commonly across the whole lifecycle of digital data, or be reused as the basis for developing more specific, content or process-aware tools.

Process level interoperability requires detailed technical specifications for the OAIS information package types, i.e. SIP, AIP and DIP. The Common Specification for Information Packages is accompanied by corresponding extending implementation specifications for E-ARK SIP, E-ARK AIP and E-ARK DIP.

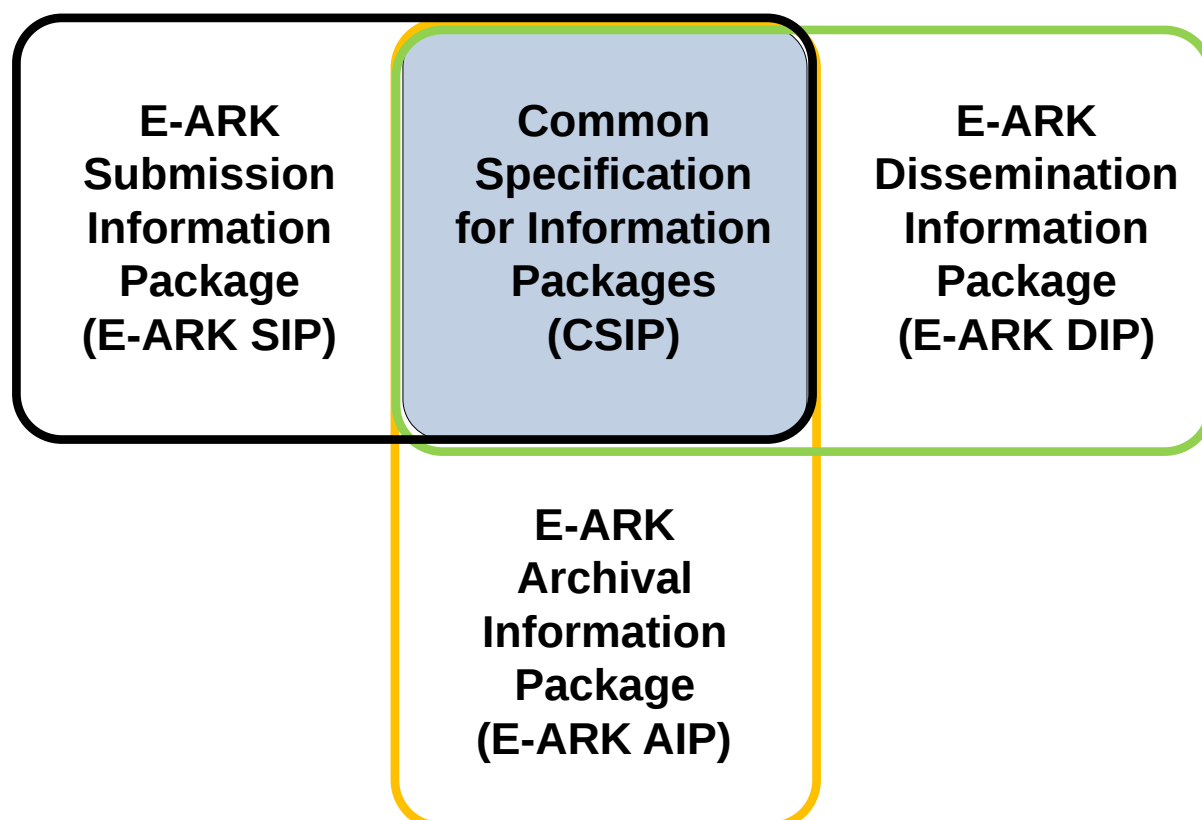


Figure 2: The scope of Common Specification for Information Packages in regard to OAIS Information Packages.

In general, the E-ARK SIP and E-ARK DIP specifications reuse and apply fully all the requirements set in this Common Specification. However, they also extend it with aspects relevant only for the respective processes (Figure 2).

For example, the E-ARK SIP specification extends the CSIP with further requirements about recording relevant information on a submission agreement and the actors of the submission process. On the other hand, the E-ARK DIP provides possibilities for describing complex access environments needed to reuse the content of a DIP.

Regarding the E-ARK AIP format, it is important to note that it does not extend the CSIP in the same way the E-ARK SIP and E-ARK DIP formats do, i.e. in the sense of a format specification inheriting all

general properties from the CSIP which is then augmented by specific AIP requirements. The reason for this is that while the SIP and the DIP are like “snapshots” in time – one capturing the state of an information package at time of submission (SIP), the other one capturing one form of delivery of the information for access (DIP) – then the AIP needs to deal with an “evolving object” which is constantly updated by preservation actions undertaken in the course of the objects life-cycle. As such, while the E-ARK AIP specification does implement all of the core metadata requirements defined in the Common Specification and extends these (for example it describes a means to record preservation actions about the IP), it does also extend the default structure of the CSIP (defined in Section 4). Essentially the AIP introduces a more complex structure which allows at the same time to securely hold an E-ARK SIP (which itself follows in full the CSIP) and at the same time add and modify additional representations over a series of preservation actions.

1.2 The Common Specification for Information Packages and Content Information Type Specifications

As an interoperability standard, it must be possible to use the CSIP regardless of the type and format of the content users need to handle. At the same time, each individual content type and file format can have specific characteristics which need to be taken into account for purposes of validation, preservation and curation.

To allow for such in-depth control over specific content types and formats, E-ARK specifications introduce the concept of Content Information Type Specifications. A Content Information Type Specification can include detailed requirements as to how content, metadata, and documentation for specific content types (for example relational databases or geospatial data) have to be handled within a CSIP (or E-ARK SIP, AIP or DIP).

As of November 2018 these Content Information Type Specifications, created by the E-ARK project and enhanced by the DILCIS Board, have been verified for usage within the Common Specification for Information Packages:

- E-ARK ERMS: Content Information Type Specification Electronic Records Management Systems. (<https://github.com/DILCISBoard/E-ARK-ERMS>) This Content Information Type Specification describes the use of the CSIP for the archiving of records exported from ERMS-type systems;
- E-ARK Geodata: This Content Information Type Specifications build upon the INSPIRE directive and describes the storing geospatial information (<https://github.com/DILCISBoard/E-ARK-Geodata>), within a CSIP compatible Information Package;
- E-ARK SIARD (SIARD1, SIARD2 and SIARDDK): This Content Information Type Specification describe the usage of the CSIP for the archiving, preservation and reuse of relational databases in one of the formats in the SIARD family (Software Independent Archiving of Relational Databases).

(<https://github.com/DILCISBoard/SIARD>) Note, that SIARD1 and SIARDDK specifications are deemed outdated by the time of writing and are only intended to be used for packaging already available SIARD1 and SIARDDK packages in a CSIP compatible Information Packages. For new occurrences of archiving relational databases the use of the SIARD2 format and according Content Information Type Specification is recommended.

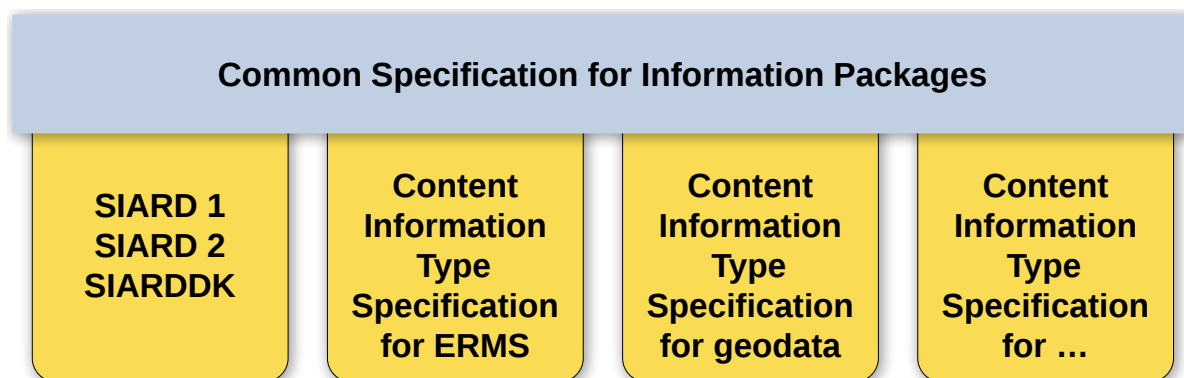


Figure 3: Common Specification for Information Packages and Content Information Type Specifications

The total number of Content Information Type Specifications is, however, unlimited and the long-term commitment of the DILCIS Board is to keep the overall environment open and inclusive. As such, interested bodies are welcome to develop their own Content Information Type Specifications, for example for 3D building projects or electronic publications. An appropriate management regime to facilitate the creation and approval of additional Content Information Type Specifications by anyone in the broader community is implemented by the DILCIS Board.

For more detailed information about the Content Information Type Specifications please look also at Section 6.1 below and check <http://www.dilcis.eu>.

1.3 Common Specification for Information Packages, OAIS Information Packages' specifications and Content Information Type Specifications

Following the discussions in the previous two Sections we can state that the overall ecosystem of E-ARK Common Specifications consists of 3-layers (Figure 4):

- The current document, the Common Specification for Information Packages, is the core which provides guidance which must be followed regardless of the process, data or lifecycle stage;
- The E-ARK SIP, AIP and DIP build on the CSIP and extend it with specific process-related aspects;
- The Content Information Type Specifications define detailed requirements for embedding and describing specific content types within a CSIP.

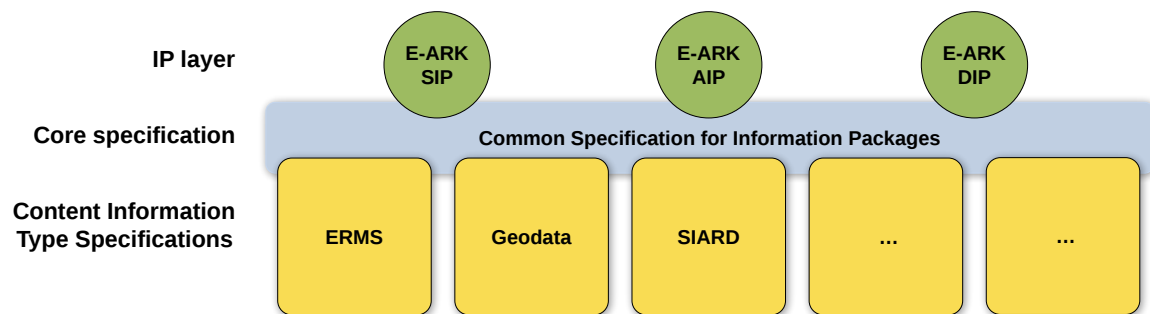


Figure 4: Relations between the Common Specification for Information Packages; E-ARK SIP, AIP and DIP specifications; and Content Information Type Specifications

Therefore the “thing encountered in the wild” is the E-ARK SIP, AIP or DIP including data according to one or many Content Information Type Specifications.

1.4. Relation to other documents

This Common Specification for Information Packages is related to the following documents:

International standards and best-practices

- Reference Model for an Open Archival Information System (OAIS), 2012, <https://public.ccsds.org/Pubs/650x0m2.pdf> This specification has used the same terminology as introduced in the OAIS model and also the same division of information package types: Submission Information Package (SIP), Archival Information Package (AIP), Dissemination Information package (DIP).
- Producer-Archive Interface Specification (PAIS) – CCSDS, 2014, <https://public.ccsds.org/Pubs/651x1b1.pdf> We have investigated the structure of a SIP presented in PAIS, but as the implementation of this specification is not very comprehensive yet (only few prototypes exist), we decided to rely mainly on the best practices introduced in other reports (see below).

Other E-ARK specifications

- E-ARK SIP Specification (<https://github.com/DILCISBoard/E-ARK-SIP>)
- E-ARK AIP Specification (<https://github.com/DILCISBoard/E-ARK-AIP>)
- E-ARK DIP Specification (<https://github.com/DILCISBoard/E-ARK-DIP>)

The E-ARK SIP, AIP and DIP specifications build on the Common Specification for Information Packages and extend it in regard to requirements derived from pre-ingest and ingest, archival storage, and access processes.

1.5. Structure of the document

The rest of this document describes the CSIP and its practical implementation. The document is divided into two logical parts.

The first part (Section 2 and Section 3) describes the generic principles of the CSIP. The main aim of these Sections is to first identify a common set of needs and thereafter present a series of requirements which an Information Package needs to follow regardless of the implementation at any given point in time:

- Section 2 provides an explanation of the need for a CSIP. The Section therefore presents some practical use cases which highlight the potential savings and increased functionality of digital archives when following internationally standardised approaches.
- Section 3 presents the core principles which need to be met in order to achieve the interoperability goal described in Section 2. Based on these requirements a set of high-level solutions are introduced regarding, for example, the structure and use of metadata within any implementation of an Information Package.

The second part of this document (Section 4, Section 5 and Section 6) presents a practical implementation of the principles described in previous Sections, as implemented according to current state-of-the-art technologies. As such, this part of the document describes the requirements which are needed to achieve practical IP interoperability:

- Section 4 presents a detailed description of the structure which must be implemented in any CSIP Information Package.
- Section 5 presents a detailed overview of metadata requirements within CSIP Information Packages with a special focus on the use of metadata elements which are needed for the automation and interoperability of archival validation and identification tasks
- Section 6 describes additional (optional) components extending the practical implementation in regard to specific aspects
 - How to create new Content Information Type Specifications
 - How to split large content objects between multiple physical IPs
 - Generic guidelines on adding (any) descriptive metadata into a CSIP Information Package

Finally, in addition to this document full examples of IPs conforming to the Common Specification for Information implementation details are available at <https://github.com/DILCISBoard/E-ARK-CSIP>.

PART I: Common Specification for Information Packages

In this part of the document we build the argument for a Common Specification for Information Packages and present the main concepts and principles for the purpose.

2 Need for establishing common ground

The vision: All digital preservation systems receive, store and provide access to information, regardless of its size, type or format, according to a set of agreed principles which allow institutions to identify, verify and validate the information in a uniform way.

The goal: Interoperability between data sources, archives and reuse environments is improved to a point where digital preservation tools can be reused across borders and institutions. This opens up new possibilities for collaboration and limits greatly the need for development resources for any single institution.

The amount of digital information being created, held and exchanged is continuously growing. This information is created with the help of numerous software tools and systems, comes in a variety of technical formats, and covers most aspects of our daily lives. Regardless of the formats and systems in question we always need to consider whether the information is needed to be retained and managed for longer periods of time. The reasons for this might be, for example:

- to meet legal and regulatory obligations
- to provide for efficient reuse
- to satisfy historical, cultural, scientific and business interest.

As of now, most tools and systems used to create information are not built for coping with long-term requirements of keeping information safe and accessible. Instead, implementations separate the short-term and long-term management of information into different systems, for example business and records systems on one hand and archival systems on the other (Figure 5).

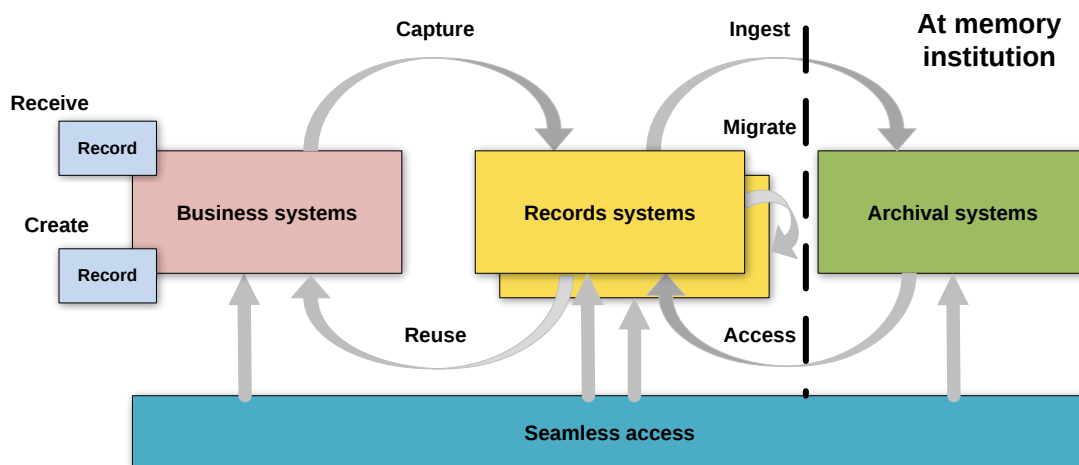


Figure 5: Information flow between live and archival systems

The implication for data owners and system managers is that information which has to be kept for extended time periods needs to be exchanged between a set of different locations, including archival systems:

- as effectively as possible,
- without endangering the authenticity and integrity of the information,
- and without limiting the possibilities for discovering and reusing the information.

As such, what we need in order to make the long-term availability of crucial information possible under (usually limited) resources is a set of principles which allow exchanging information in a common way across the systems participating in archival workflows and processes, i.e. create a set of interoperability specifications. For archival information packages we have identified the following interoperability scenarios (Figure 6):

- Export of data and metadata from source systems and transfer to SIP creation tools (or directly, as an SIP, into preservation systems);
- Transfer of SIPs from SIP creation tools to preservation systems;
- Exchange of preservation system where all AIPs need to be transferred (ingested) into a new technological platform;
- Distributed storage and synchronisation of AIPs between multiple (technologically different) preservation systems;
- Exchange of DIPs between preservation systems and access platforms or portals;
- Exchange of DIPs between various access platforms of portals.

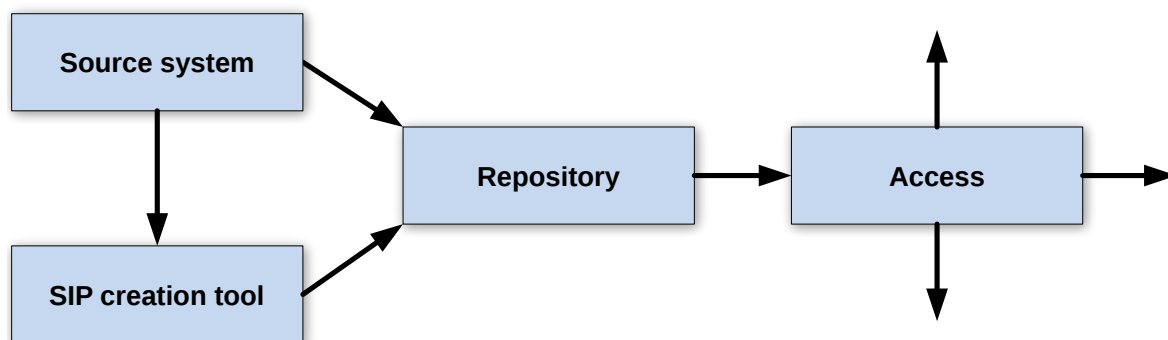


Figure 6: Archival workflow and tool ecosystem

As of 2014 (the start of the development of this specification) the state of interoperability in digital preservation was rather poor. While national and institutional practical implementation-level specifications existed to serve the need for data and metadata packaging and exchange, these were by large not interoperable with each other. On the contrary, available and widely used international specifications (most notably METS and PREMIS) lack the necessary implementation-level detail, needed in order to serve as an authoritative source for practical interoperability.

This situation has a remarkable effect on the cost of digital preservation. Namely, the tools developed in individual institutions are not reusable across institutional and state borders and therefore need to be redeveloped at each single location. Globally, this raises the cost of digital preservation to a level which makes it not affordable for smaller institutions and, at the same time, does often not allow developing tools which would be sufficiently mature, user-friendly and prone to errors. As well, the multitude of national or institutional specifications does not allow internationally active source system providers (e.g. Oracle, Microsoft) to build a single native archiving functionality into their products, meaning that there is a need for bespoke development (and therefore added cost) for each installation of these source systems across all sectors and countries.

To overcome these limitations this document proposes a universal common specification, which can be implemented across borders, for how data and metadata should be structured and packaged when transferred to archival systems, ingested and preserved in these, and re-used. Such a specification will allow data owners to build standardised interfaces for the export of their data regardless of the archives in question; and digital archives to build standardised interfaces for data ingest and access, regardless of the data providers and users in question.

Further, the aim of the common specification is to be sufficiently detailed and technical to allow for extended collaboration in regard to software development and pooling. Ideally the tools which implement the common specification for data export, transfer, ingest, preservation and reuse are exchangeable between institutions and administrations with minimal effort. This in turn shall lead to a significant decrease in resources needed from any single institution and at the same time opens up

an extended market for commercial software providers.

3 Principles for interoperable Information Packages

At the heart of any standardisation activity has to be a clear understanding of the needs and aims which have to be addressed. This is also the goal of this Section, which presents a series of high level principles to guide the technical details delivered in Part II of this specification.

Most of the principles are driven by the aim of interoperability –Information Packages shall be easy to exchange, identify, validate and (re)use with a wide variety of software tools and systems.

Another crucial factor to take into account is long-term sustainability. Practical technical and semantic interoperability is possible only when a certain set of technologies have been agreed upon and implemented. However, any technology will become outdated sooner or later and previously agreed-upon approaches have to be updated to accommodate new, better and more efficient technologies and standards. Because of this, the developers of this Common Specification for Information Packages have reused, as much as possible, existing powerful, standardised and well-established best practices for the technical implementation of an Information Package (Part II of this document). This does not mean that the technical implementation details will not need to be changed in future, only that the need will arise later rather than sooner. To achieve long-term sustainability of the Common Specification for Information Packages, we present below a set of generic principles which must be followed when updating any specific implementation details at any point in time.

The principles present a conceptual view of an Information Package, including an overall IP data model, and use of data and metadata. An implementation of this conceptual view is presented later, in Part II of this document.

Each principle has a sequential number and a short description. The description includes always a MoSCoW (MUST/MUST NOT, SHOULD/SHOULD NOT, COULD, WOULD) prioritisation statement. The short description of each principle is followed by a rationale which describes the reason and background for the principle.

3.1 General principles

Principle 1.1

It MUST be possible to include any data or metadata in a Information Package regardless of its type or format.

This is one of the most crucial principles of the CSIP. In order to be truly “common”, technical implementations of the CSIP MUST NOT introduce limitations or restrictions which are only applicable to certain data or metadata types. If an Information Package implementation fails to meet this principle it is not possible to use it across different sectors and tools, thereby limiting practical interoperability.

Principle 1.2:

The Information Package MUST NOT restrict the means, methods or tools for exchanging it.

Tools and methods for transferring Information Packages between locations are constantly evolving. It is also possible that different methods are preferred for packages of varying sizes. In order to achieve that a CSIP Information Package is truly interoperable across different platforms it therefore MUST NOT introduce limitations or restrictions which would be impossible to be met by specific information exchange tools or channels.

As such the CSIP does also not define the principle to use a particular transfer package or envelope. The scope of the CSIP is limited to the structure and requirements for data and metadata within the package. Different implementers are welcome to choose their own methods on top of the CSIP.

Principle 1.3

The package format MUST NOT define the scope of data and metadata which constitutes an Information Package.

One of the fundamental principles of the CSIP is that it MUST allow each individual repository to define the (intellectual) scope of an Information Package and its relations to real life entities. As such, any implementation of the CSIP MUST be equally usable for packaging, for example, the entire content of an ERMS as an single IP; or for extracting each record and its metadata from the ERMS individually and packaging each as a separate IP.

Out of the previous we can also derive that a CSIP specification MUST NOT define that, for example, a SIP should conform to exactly one AIP. Instead the CSIP MUST allow for the inclusion of “anything that the implementer wants to define as a SIP, AIP or DIP” and allow for “any relationships (1-1; 1-n; n-1; n-m) between SIPs, AIPs and DIPs”.

Principle 1.4:

The Information Package SHOULD be scalable.

One of the practical concerns for Information Packages is their size. Many digital repositories have problems with data objects and metadata of increasing sizes, making it especially difficult to carry

out tasks related to data or metadata validation, and identification and modification. For example, Information Packages including relational databases or born-digital 3D movies can easily reach TB sizes.

Consequently, any current or future implementation of the CSIP is required to provide for appropriate scalability mechanisms (for example: mechanisms for splitting large-scale data or metadata).

Principle 1.5:

The Information Package MUST be machine-readable

To support the goal of automating ingest, preservation and access workflows each of the implementations of the CSIP must be machine-actionable. This means that decisions about the use of metadata syntax and semantics as well as the physical structure must be expressed explicitly and in a clear way. This, in turn, allows the specification to be implemented in the same way across different tools and environments.

Principle 1.6:

The Information Package SHOULD be human-readable

In long-term preservation we also need to take into account that “forgotten” Information Packages might be found long after details about the implementation are gone and no tools to access the package are available. For these scenarios it is crucial to ensure that the structure and metadata of the Information Package are understandable with minimal effort by using simple tools like text editors and file viewers.

In practice this means that any implementation of the CSIP should ensure that folder and file naming conventions allow for the human identification of package components, and that the semantics of the package is explicit.

Principle 1.7:

The Information Package MUST NOT prescribe the use of a specific preservation method

Different preservation institutions and different types of data need to use different methods for long-term preservation; migration and emulation are typical examples. A CSIP implementation MUST NOT prescribe the use of a specific preservation method. Instead it should allow users to document and/or add arbitrary data or metadata as necessary for preservation purposes.

3.2 Identification of the Information Package

Principle 2.1:

The Information Package OAI type (SIP, AIP or DIP) MUST be clearly indicated.

One of the first tasks in analysing any Information Package is to identify its current status in the overall archival process. Therefore, any Information Package must explicitly and uniformly identify itself as a SIP, AIP or DIP.

Principle 2.2:

Any Information Package MUST clearly identify the Content Information Type(s) of its data and metadata.

As stated in Principle 1.1 any Information Package MUST be able to include any kind of data and metadata. At the same time we have introduced in earlier Sections the concept of Content Information Types which allow users to achieve more detailed control and fine-grained interoperability. As such, any CSIP Information Package MUST include a statement about which Content Information Type Specification(s) has been followed within the Information Package, or on the contrary, indicate clearly that no specific Content Information Type Specification has been followed.

The practical implication of principles 1.1, 2.1 and 2.2 is that, once these have been followed in implementations, we can in fact develop modular identification and validation tools and workflows. While generic components can carry out high level tasks regardless of the Content Information Type, it is possible to detect automatically which additional content-aware modules need to be executed.

Principle 2.3:

Any Information Package MUST have an identifier which is unique and persistent within the repository.

In order to manage a digital repository and provide access services each Information Package stored in the repository MUST be identified uniquely at least within the repository. At the same time a CSIP implementation MUST NOT limit the choice of the exact identification mechanism, as long as the mechanism is implemented consistently throughout the repository.

Principle 2.4:

Any Information Package SHOULD have an identifier which is globally unique and persistent.

In addition to the previous principle, it is recommended that the identification mechanism used at the repository provides for global uniqueness and persistence of Information Package IDs. The

application of globally unique and persistent identifiers allows repositories to participate more easily in cross-institutional information exchange and reuse scenarios (for example participation in national or international portals, or cross-repository duplication of AIP preservation). However, the CSIP MUST NOT limit the choice of the exact identification mechanism.

Principle 2.5:

All components of an Information Package MUST have an identifier which is unique and persistent within the repository.

As stated above, a Information Package MUST be flexible enough to allow for the inclusion of any data or metadata depending on the needs of the repository and its users. As well, an Information Package might include additional support documentation like metadata schemas, user guidelines, contextual documentation etc. Regardless of which and how many components constitute a full Information Package, all components MUST have a unique and persistent identifier which allows for the appropriate linking of data, metadata and all other components. This, in turn, is one of the most crucial aspects towards achieving an interoperable way towards maintaining package integrity.

It is also worth mentioning that in any implementation it is only necessary to achieve identifier uniqueness and persistence within an individual Information Package. If this is the case, repository-wide uniqueness is easily achieved when combining the package ID (unique according to principle 2.3) and the component ID.

The components of a Information Package are explained in more detail in the following section.

3.3 Structure of the Information Package

Principle 3.1:

The Information Package MUST ensure that data and metadata are logically separated from one another.

At it's highest level, each Information Package can be logically sub-divided into data and metadata. This logical separation minimises the effort required to identify or validate content/metadata and simplifies long term preservation actions. For example, ingest tools may implement separate, efficient metadata identification and validation tasks as opposed to content format identification and normalisation tasks. Over a package's lifetime this separation also facilitates long-term preservation tasks, such as partial metadata or data updates that don't put the package's integrity at risk. Regardless of a package's physical structure, the Information Package MUST provide logical separation of data and metadata in the package's manifest.

Principle 3.2:

The Information Package SHOULD ensure that data and metadata are physically separated from one another.

In addition to the logical separation of components it is beneficial to have data and metadata physically separated (i.e. formatted as individual computer files or clearly separated bitstreams). This allows digital preservation tools and systems to update respective data or metadata portions of an Information Package without endangering the integrity of the whole package.

Principle 3.3:

The structure of the Information Package SHOULD allow for the separation of different types of metadata

In addition to the previous principle it is recommended to explicitly divide metadata into more specific components. While the definitions of metadata types vary a lot between implementations it is our recommendation to divide metadata logically and physically at least into descriptive and preservation metadata.

Principle 3.4:

The structure of the Information Package MUST allow for the creation of data and metadata in multiple representations.

The concept of representations is one of the fundamental building blocks in digital preservation. As technologies evolve and get obsolete, data and metadata is constantly updated in order to ensure long-term accessibility, therefore creating new versions or representations of the data and metadata.

Expressing representations within the logical and physical structure of an Information Package helps institutions to explicitly understand the various states of the information throughout its lifecycle, therefore improving also the ease of long-term management and reuse of the information.

Principle 3.5:

The structure of the Information Package MUST explicitly define the possibilities for adding additional components into the Information Package.

In addition to data and metadata, institutions might have the need to include additional components in an Information Package. For example, implementers might decide that XML Schemas about metadata structures and additional binary documentation about the original IT environment have to be added to the package.

If this is the case, the CSIP Information Package MUST NOT limit which components can constitute an Information Package, and MUST offer clearly defined extension points for the inclusion of these additional components into the Information Package. At the same time these extension points MUST be defined in a way which does not interfere with other components (i.e. the extension points MUST be clearly separated from other components of an Information Package).

Principle 3.6:

The Information Package SHOULD follow a common conceptual structure regardless of its technical implementation.

Based on principles 3.1 – 3.5 we present a common structure for any CSIP Information Package (Figure 7).

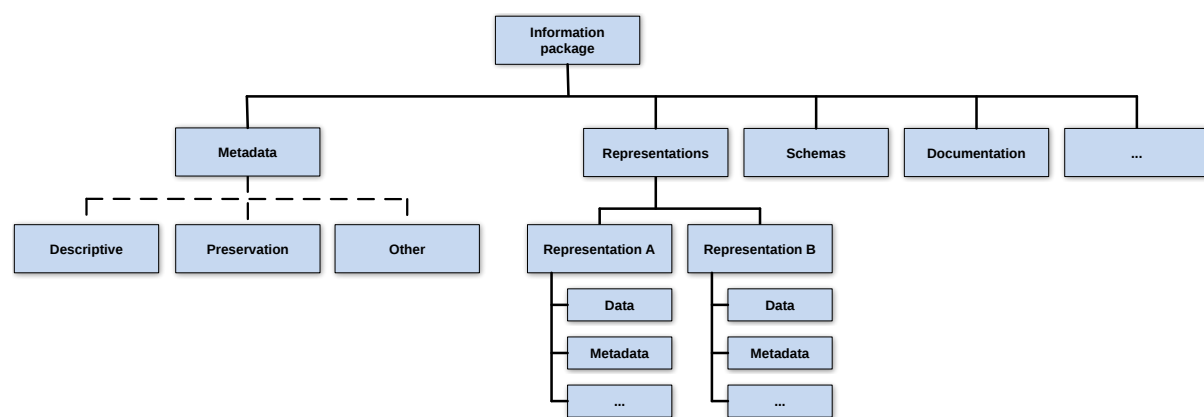


Figure 7: Conceptual structure of the Common Specification

Following Principle 3.1 the package MUST include a high-level structural component for metadata which includes at least relevant metadata for the whole package. In addition the representations MUST internally separate between data and metadata (though note that the CSIP does not mandate that both data and metadata must be available in all representations).

Following Principle 3.2 we strongly recommend this logical structure to be manifested as a physical folder structure.

Following Principle 3.3 we strongly recommend sub-dividing any package metadata into separate metadata sub-types.

Following Principle 3.4 the structure separates explicitly the representations of data and metadata into a separate structural component.

Following Principle 3.5 repositories and their users have the possibility to add any additional components (as an example for schemas and binary support documentation) either as extensions to the whole Information Package or into a specific representation.

This common structure SHOULD be followed throughout all implementations of the CSIP.

The conceptual structure presented above can be implemented in various ways – for example the components might be defined by accompanying metadata or explicitly through a physical structure. However, it is not reasonable to have multiple implementations available at once as this would lead to unnecessary complexity in developing interoperable tools for creating, processing and managing Information Packages. At the time being for CSIP the highly recommended implementation is to use a fixed physical folder structure in combination with a manifest in the form of a METS-document (see Section 4 and Section 5) as the implementation of this specification.

At the same time it is clear that any given technical implementation will become obsolete in time, for example as new transfer methods and storage solutions emerge. As such this specification does not prohibit the take-up of any emerging logical or physical technical solutions.

3.4 Information Package Metadata

Principle 4.1:

Metadata in the Information Package MUST conform to a standard.

In order to exchange, validate, process and reuse Information Packages in an interoperable and automated way we need to standardise how crucial metadata are presented in the package. “Crucial metadata”, is defined in this specification as the core information about how the package content has been created and managed (administrative and preservation metadata), explicit descriptions about the structure of the package (structural metadata) and the technical details of the data themselves (technical metadata).

In order to ensure that these metadata are understood and implemented in a common and interoperable way in any Information Package, the use of established and widely used metadata standards is highly recommended. In the current implementation a large proportion of such metadata is covered by the widely used METS and PREMIS standards (see Section 5).

Principle 4.2:

Metadata in the Information Package MUST allow for unambiguous use.

Many metadata standards support multiple options for describing specific details of an Information Package. However, such interpretation possibilities can also lead to different implementations and ultimately to the loss of interoperability.

To overcome this risk the CSIP requires that, while developing a specific implementation, the chosen metadata standard **MUST** be reviewed in regard to potential ambiguity. If needed, the selected metadata standard **MUST** be further refined to meet the needs of interoperability and automation.

Principle 4.3:

*The Information Package **MUST NOT** restrict the addition of supplementary metadata.*

Previous principles state the importance of controlled metadata for interoperability purposes. At the same time the opposite applies for other types of metadata, most prominently for resource discovery (also called descriptive) or Content Information Type specific technical and structural metadata. In order to not limit the widespread adoption of the CSIP it has to be possible for any implementer to add any metadata next to the mandatory metadata components needed for package level automation and interoperability.

In case organisations need to prescribe further details about descriptive or Content Information Type specific metadata for a deeper level of interoperability it is possible to use the mechanism of Content Information Type Specifications described above.

To summarise the requirements above from a more technical perspective, the CSIP foresees a modular approach towards Information Package metadata:

- All Information Packages share a common core of metadata which allows for the common development of high-level package creation, validation, identification and reuse tools;
- The rest of the metadata in the Information Package might follow additional agreements which have been made in order to develop specific tools such as, for example, tools to manage archival descriptions in EAD, or for specific Content Information Types like relational databases in the SIARD2 format.

PART II: Implementation of the CSIP

In this part of the document we present an implementation of the requirements and principles presented in Part I of the specification for CSIP. The implementation consists of two core elements: a fixed physical structure of a CSIP Information Package (Section 4) and the exact use of metadata using the “Metadata Encoding & Transmission Standard” (METS) <http://www.loc.gov/standards/mets/> and “PREservation Metadata Implementation Strategies” (PREMIS) <http://www.loc.gov/standards/premis/> format (Section 5).

As explained earlier, any implementation using a metadata standard will inevitably become obsolete. However, the authors have reused available best practices and established standards, and held

discussions with the digital preservation community to ensure that the implementation is as future proof as possible.

4. CSIP structure

The preferred implementation of the logical model described in Principle 3.6 is a strict physical (folder) structure precisely following the logical structure. While the specification doesn't prohibit alternative implementations of the conceptual model, the practise is not recommended.

The main reason for this implementation decision is that a fixed and documented folder structure makes the package layout clear to both human users and automated tools. The main benefit this clarity is that many archival tasks, e.g. file format risk analysis, can be executed directly on the data portion of the package structure, as opposed to first processing potentially large amounts of metadata for file locations. This allows for more efficient processing which is valuable in the case of large collections and bulk operations. In short, we believe that a fixed folder structure provides efficiency and scalability.

The authors are aware that many data storage solutions do not explicitly support folder structures, but use other means for structuring and storing AIP data and metadata. However, the purpose of this specification is to facilitate and support Information Package interoperability. When storage solution don't support implementation of the package structure for native AIP storage, it is still possible to implement the physical structure for SIPs and DIPs. While systems need to implement transformations between Common Specification IPs and internal AIPs it allows interoperability between tools that support the common specification, easy transfer of IPs to new repository systems or storage solutions and the establishment of multi-repository duplicated storage solutions.

4.1. Folder structure of the CSIP

The CSIP folder structure is presented in Figure 8 below. The structure directly follows the principles of the conceptual data model, dividing the package components into individual folders for representations, metadata, and other components. All of folders described ought to be present even if they are empty.

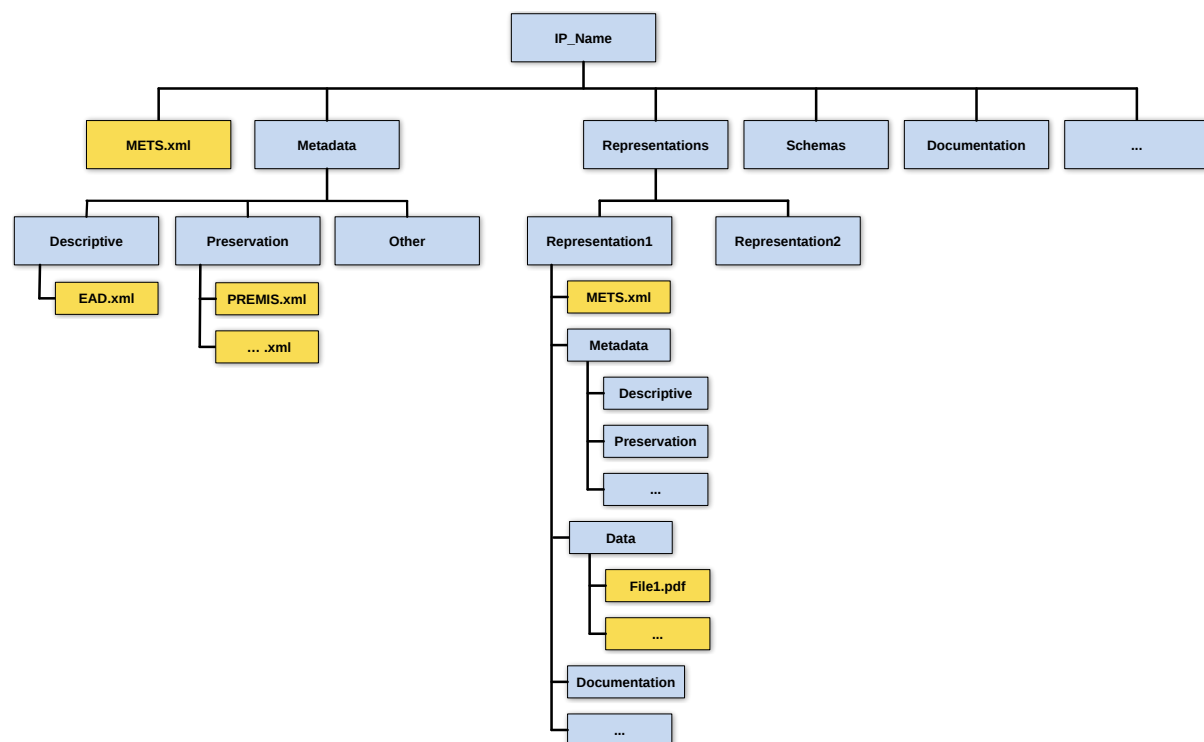


Figure 8: CSIP Information Package folder structure

The implementation requirements of the CSIP Information Package structure are:

CSIPSTR1: Any Information Package MUST be included within a single physical root folder (known as the “Information Package root folder”). For packages presented in an archive format, see CSIPSTR3, the archive MUST unpack to a single root folder.

CSIPSTR2: The Information Package root folder SHOULD be named with the ID or name of the Information Package, that is the value of the package METS.xml’s root `<mets>` element’s `@OBJID` attribute.

CSIPSTR3: The Information Package root folder MAY be compressed (for example by using TAR or ZIP). Which specific compression format to use needs to be stated in the Submission Agreement.

CSIPSTR4: The Information Package root folder MUST include a file named `METS.xml`. This file MUST contain metadata that identifies the package, provides a high-level package description, and describes its structure, including pointers to constituent representations.

CSIPSTR5: The Information Package root folder SHOULD include a folder named `metadata`, which SHOULD include metadata relevant to the whole package.

CSIPSTR6: If preservation metadata are available, they SHOULD be included in sub-folder `preservation`.

CSIPSTR7: If descriptive metadata are available, they SHOULD be included in sub-folder `descriptive`.

.

CSIPSTR8: If any other metadata are available, they MAY be included in separate sub-folders, for example an additional folder named `other`.

CSIPSTR9: The Information Package folder SHOULD include a folder named `representations`.

CSIPSTR10: The `representations` folder SHOULD include a sub-folder for each individual representation (i.e. the “representation folder”). Each representation folder should have a string name that is unique within the package scope. For example the name of the representation and/or its creation date might be good candidates as a representation sub-folder name.

CSIPSTR11: The representation folder SHOULD include a sub-folder named `data` which MAY include all data constituting the representation.

CSIPSTR12: The representation folder SHOULD include a metadata file named `METS.xml` which includes information about the identity and structure of the representation and its components. The recommended best practice is to always have a `METS.xml` in the representation folder.

CSIPSTR13: The representation folder SHOULD include a sub-folder named `metadata` which MAY include all metadata about the specific representation.

CSIPSTR14: The Information Package MAY be extended with additional sub-folders.

CSIPSTR15: We recommend including all XML schema documents for any structured metadata within package. These schema documents SHOULD be placed in a sub-folder called `schemas` within the Information Package root folder and/or the representation folder.

CSIPSTR16: We recommend including any supplementary documentation for the package or a specific representation within the package. Supplementary documentation SHOULD be placed in a sub-folder called `documentation` within the Information Package root folder and/or the representation folder.

4.2. Implementing the structure

The requirements presented in Section 4.1 leave scope for decisions during the implementation. For clarity we provide two extreme examples – both the simplest and the most complete use of the structure.

In the simple case the structure can be implemented for the most part by adhering to only the MUST requirements. An example of this is shown in Figure 9.

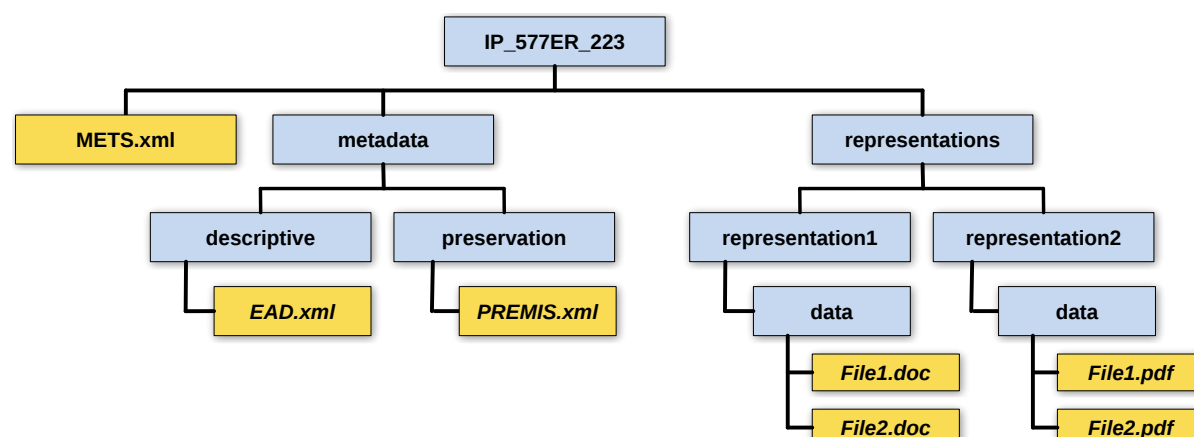


Figure 9: A simple CSIP structure example.

Note that the representations have been kept as simple as possible. All metadata for both the package and the representations (METS, EAD and PREMIS metadata) are located in the Information Package folder rather than in the representation folders.

Such a simple implementation is reasonable in scenarios where the amount of data and metadata is limited. However, in the case of large Information Packages (e.g. packages comprising several representations, each with 1,000,000 files) the size of both the `METS.xml` file and preservation metadata can become too large to manage efficiently. In such large data scenarios it might prove necessary to implement all the features of the structure presented in the previous Section.

An example of a full implementation is illustrated in Figure 10. One observation is that the representations of the complete example essentially repeat the simple structure. Structural and preservation metadata in METS and PREMIS formats are available in both the Information Package folder, package level metadata, and the representation folders, representation level metadata. This illustrates how the full structure facilitates the management of single representations bringing benefits such as simpler metadata versioning. It is worth noting, to avoid confusion, that we recommend adoption of a common approach, either adding metadata to representations or not. That is, we recommend placing all representation-relevant metadata in a single location. You may use either the package level metadata folder or the representation level metadata folder, but do not divide metadata between the two inconsistently, with representation metadata stored at both package and representation level. We also do not recommend duplication of metadata or content of optional folders, e.g. schemas, documentation, etc., between the Information Package folder and representation folders.

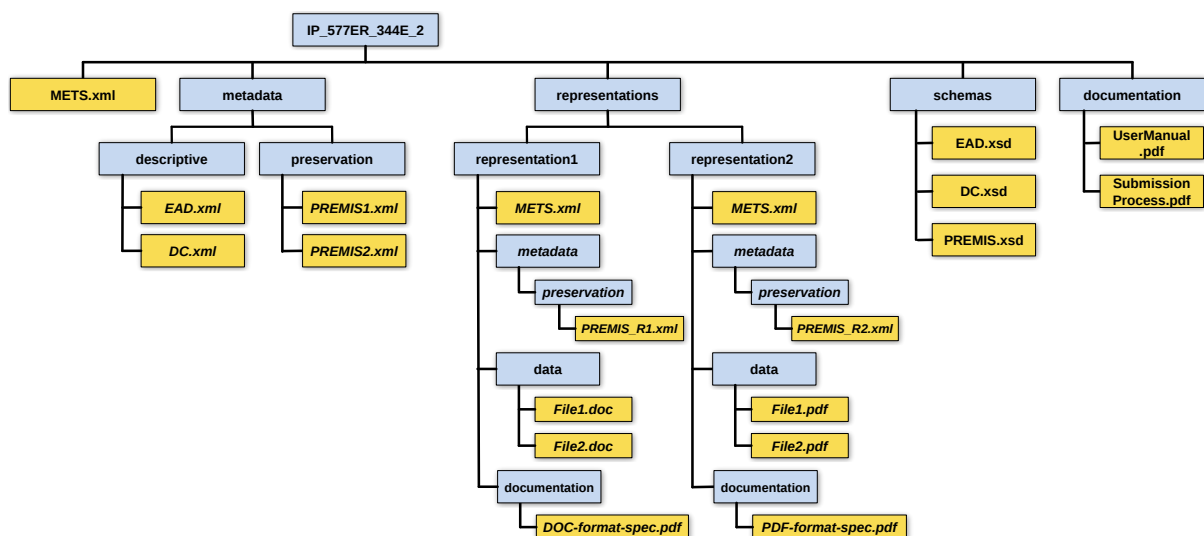


Figure 10: Example of the full use of the CSIP structure

5. Use of metadata

5.1. General requirements for metadata in a CSIP Information Package

The primary consideration with metadata is, as with the rest of this specification, interoperability. Specifically the high-level technical interoperability and tasks that allow an Information Package to be prepared, transferred and received, regardless of the institutions or tools involved. These tasks include:

- Identifying uniquely an Information Package and its components;
- Validating an Information Package and its contents;
- Establishing the authenticity of the Information Package;
- Accessing the contents of an Information Package.

In technical terms, the CSIP tries to control metadata needed by tools or users to:

- navigate data and metadata components within the package (i.e. packaging metadata);
- to validate that no component has been damaged during transfer or preservation (i.e. fixity information);
- to understand the processes used when creating and managing the package (i.e. provenance and preservation metadata); and
- to understand how the data within the package can be accessed (i.e. representation information).

We regard descriptive metadata and most detailed technical metadata as outside the scope of the CSIP. As such, the CSIP does not aim to provide detailed semantic interoperability between different

systems. However, as noted in Section 1.2, implementers are welcome to use the Content Information Type Specifications to achieve interoperability at a more detailed level.

We describe the core metadata requirements using METS (Metadata Encoding & Transmission Standard, <http://www.loc.gov/standards/mets/>). We describe the core elements used, METS describes other available elements which may be used in addition to those detailed in the CSIP.

Some metadata requirements are implied by the structure presented in the previous Section. As seen, one or more METS files can be present in a package. The METS file describing the whole package is from now called the “Package METS” and the METS file in the Representation folders are called “Representation METS”. The detailed specification of using METS within the CSIP is available in Section 5.3.

In addition to the METS files the CSIP recommends the inclusion of PREMIS metadata (PREservation Metadata Implementation Strategies, <http://www.loc.gov/standards/premis/>), placed in appropriate preservation metadata folders. This facilitates interoperable approach towards provenance and access to Information Packages. However, we recognise that, especially in the case of SIPs, appropriate preservation metadata is not always available. As such this is also not an absolute requirement though highly desirable. A specification of the use of PREMIS within the CSIP is available in Section 5.4.

The specification does not prohibit the use of additional metadata within Information Packages.

5.2 General requirements for the use of metadata

Before detailing the requirements covering the use of METS and PREMIS, we would like to highlight some general metadata concerns which should be handled consistently.

The use of identifiers

Both METS, and by derivation the CSIP, make extensive use of the XML ID data type (see: <https://www.w3.org/TR/xml-id/>). Valid XML IDs must also conform to the NCName restrictions (see: <https://www.w3.org/TR/REC-xml-names/#NT-NCName>), and must begin a letter or an underscore character ('_'), and contain no characters other than letters, digits, hyphens, underscores, full stops, plus some extension and combination characters.

If your package makes use of IDs that don't conform to this specification one mitigating strategy is to add a prefix to all identifiers. For example UUIDs are a common choice for arbitrary identifiers and comprise of hex characters and hyphens, e.g. the value 906F4F12-BA52-4779-AE2C-178F9206111F. NCName and xml:id values can't legally begin with numeric characters making many UUIDs, including the example, invalid. The examples below show two possible solutions, using prefixes:

Example 1: The prefix consists of the identifier type acronym and a hyphen: `uuid-`.

```
<dmdSec ID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F" CREATED="2018-04-24T14:37:49.609+01:00">
```

Example 2: A generic prefix:ID.

```
<dmdSec ID="ID906F4F12-BA52-4779-AE2C-178F9206111F" CREATED="2018-04-24T14:37:49.609+01:00">
```

Note that the identifiers specified within the CSIP are mainly used as internal references between Information Package components. Prefixes are not mandatory, but if they are used, we recommend choosing a single prefix and using it consistently across all IDs in the package.

Referencing between files within a CSIP Information Package

The CSIP strongly recommends that all components of an information package (i.e. all data, metadata and other parts) are stored as discrete files within the package. While this approach simplifies the management of the Information Package, making it easier to aggregate, validate and modify the package, it also necessitates a clear method for recording and resolving referencing between files.

For example, when using the CSIP referencing can occur between:

- descriptive and administrative metadata files described in the `amdSec` and `dmdSec` elements;
- content (data) files, components of documentation, and schemas described in the `fileSec` element;
- representation METS files described in the Package METS file's `fileSec` and `structMap` elements.

A consistent approach to referencing between package components is a fundamental requirement for Information Package validation and integrity checking. Different technical solutions exist for referencing and resolving but they're not universally supported by all digital preservation tools. To ensure interoperability, all references within a CSIP Information Package must adhere the requirements stated in this specification.

Referencing other packages

It is important that external references, i.e. to other packages, are expressed consistently, in the same manner as internal references. All references to other packages MUST USE the `mets/@OBJID` value of the target package.

Examples

This document illustrates use of the specification using generic examples. These examples use values from vocabularies and elements defined by the CSIP to demonstrate some specific uses of the specification. Outside these CSIP specific values the METS values are fictive and are not meant to constitute real life examples. More complete examples will be provided in the CSIP user guidelines that will be published at a future date.

5.3 Use of METS

The main requirement for METS files in a CSIP Information Package is that these need to follow the official METS Schema version 1.12 <http://www.loc.gov/standards/mets/mets-schemadocs.html> (by CSIP used version in May 2019) and the extension schema developed for the CSIP and published by the DILCIS Board. As new versions of METS Schema become available the DILCIS Board will evaluate these and, if necessary, update the CSIP respectively.

The following text assumes knowledge of the principles of the METS specification. If this is not the case, please consult the official documentation before continuing.

The METS specification requires a METS profile document which describes the use of METS and the METS elements. All the requirements described in this specification is also expressed with a METS profile for the CSIP <https://github.com/DILCISBoard/E-ARK-CSIP/tree/master/profile>.

METS allows metadata to be both embedded and referenced. The CSIP itself allows both the embedding of metadata within the `METS.xml` file but note that for scalability concerns the CSIP only recommends the use of referencing. This means that the CSIP only describes referencing of metadata.

The rest of this Section describes the use of each of the METS elements: `mets`, `metsHdr`, `dmdSec`, `amdSec`, `fileSec` and `structMap`. For each element we concisely explain restrictions imposed by the CSIP on top of those in the official METS documentation. A CSIP implementation can choose to extend the restrictions in line with requirements specific to the implementation. In these cases follow the METS documentation and create an implementation which uses the CSIP as a base profile.

Differences between creating a package METS file and representation METS file are described when relevant.

All names of elements and attributes below are expressed using the XPath notation (i.e. `element/sub-element/@attribute`)

5.3.1. Use of the METS root element (element mets)

The METS document's root element (`<mets>`) describes the container for the information being stored and/or transmitted. The `<mets>` element of a CSIP conformant METS document uses attributes from the METS specification and additional attributes defined in the CSIP extension schema.

As well as the METS and CSIP extension attributes, the METS document's root `<mets>` element must define all of the relevant namespaces and locations for XML schema used in the package. This is done using the `@xmlns` and `@xsi:schemaLocation` attributes as described in "Referencing a Schema in an XML Document". The schema identifiers and locations for a typical CSIP `<mets>` element are shown below:

```
<mets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns="http://www.loc.gov/METS/"
      xmlns:csip="https://DILCIS.eu/XML/METS/CSIPExtensionMETS"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance schemas/
                          XMLSchema.xsd http://www.loc.gov/METS/ schemas/mets.xsd http://www.w3
                          .org/1999/xlink schemas/xlink.xsd https://DILCIS.eu/XML/METS/
                          CSIPExtensionMETS schemas/CSIPExtensionMETS.xsd" ... >
```

Location of XML schema

When using XML schemas the availability of the actual schema resources should be considered. Externally hosted resources are not guaranteed to be available in the future, or in restricted operating environments. We recommend that copies of all XML schema resources should be included in the information package, located in appropriate "schemas" folders at package or representation level. When schemas have been included in the package `schemas` folder, links to the schema documents should refer to the relative path of the schema file within the package, i.e. `schemas/mets.xsd`.

The specific requirements for the root element and its attributes are described in the following table.

ID	Name & Loc	Description & usage	Card & Level
CSIP1	Package Identifier <code> mets/@OBJID</code>	The <code> mets/@OBJID</code> attribute is mandatory, its value is a string identifier for the METS document. For the package METS document, this should be the name/ID of the package, i.e. the name of the package root folder. For a representation level METS document this value records the name/ID of the representation, i.e. the name of the top-level representation folder.	1..1 MUST
CSIP2	Content Category <code> mets/@TYPE</code>	The <code> mets/@TYPE</code> attribute MUST be used to declare the category of the content held in the package, e.g. book, journal, stereograph, video, etc.. Legal values are defined in a fixed vocabulary. When the content category used falls outside of the defined vocabulary the <code> mets/@TYPE</code> value must be set to “OTHER” and the specific value declared in <code> mets/@csip:OTHERTYPE</code> . The vocabulary will develop under the curation of the DILCIS Board as additional content information type specifications are produced. See also: Content Category	1..1 MUST
CSIP3	Other Content Category <code> mets[@TYPE='OTHER']/@csip:OTHERTYPE</code>	When the <code> mets/@TYPE</code> attribute has the value “OTHER” the <code> mets/@csip:OTHERTYPE</code> attribute MUST be used to declare the content category of the package/representation. See also: Content Category	0..1 SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP4	Content Information Type Specification <code>mets/@csip:CONTENTINFORMATIONTYPE</code>	Used to declare the Content Information Type Specification used when creating the package. Legal values are defined in a fixed vocabulary. The attribute is mandatory for representation level METS documents. The vocabulary will evolve under the care of the DILCIS Board as additional Content Information Type Specifications are developed. See also: Content information type specification	0..1 SHOULD
CSIP5	Other Content Information Type Specification <code>mets[@csip:CONTENTINFORMATIONTYPE='OTHER']/@csip:OTHERCONTENTINFO</code>	When the <code>mets/@csip:CONTENTINFORMATIONTYPE</code> has the value "OTHER" the <code>mets/@csip:OTHERCONTENTINFORMATIONTYPE</code> must state the content information type.	0..1 MAY
CSIP6	METS Profile <code>mets/@PROFILE</code>	The URL of the METS profile that the information package conforms with.	1..1 MUST

Example: METS root element showing use of `csip:@OTHERTYPE` attribute when an appropriate package content category value is not available in the vocabulary. The `@TYPE` attribute value is set to OTHER.

```
<mets:mets OBJID="uuid-4422c185-5407-4918-83b1-7abfa77de182" LABEL="Sample
  CSIP Information Package" TYPE="OTHER" OTHERTYPE="Patterns" PROFILE="
  https://earkcsip.dilcis.eu/profile/E-ARK-CSIP.xml" schemaLocation="http
  ://www.loc.gov/METS/ http://www.loc.gov/standards/mets/mets.xsd http://
  www.w3.org/1999/xlink http://www.loc.gov/standards/mets/xlink.xsd https
  ://dilcis.eu/XML/METS/CSIPExtensionMETS https://dilcis.eu/XML/METS/
  CSIPExtensionMETS/DILCISExtensionMETS.xsd">
</mets:mets>
```

Example: METS root element illustrating the use of a custom `csip:@OTHERCONTENTINFORMATIONTYPE` attribute value when the correct content type value does not exist in the vocabulary. The `csip:@CONTENTINFORMATIONTYPE` attribute value is set to OTHER.

```
<mets:mets OBJID="uuid-4422c185-5407-4918-83b1-7abfa77de182" LABEL="Sample
  CSIP Information Package" TYPE="Datasets" CONTENTINFORMATIONTYPE="
  OTHER" OTHERCONTENTINFORMATIONTYPE="FGS Personal, version 1" PROFILE="
  https://earkcsip.dilcis.eu/profile/E-ARK-CSIP.xml" schemaLocation="http
  ://www.loc.gov/METS/ http://www.loc.gov/standards/mets/mets.xsd http://
  www.w3.org/1999/xlink http://www.loc.gov/standards/mets/xlink.xsd https
  ://dilcis.eu/XML/METS/CSIPExtensionMETS https://dilcis.eu/XML/METS/
  CSIPExtensionMETS/DILCISExtensionMETS.xsd">
</mets:mets>
```

5.3.2. Use of the METS header (element metsHdr)

The purpose of the METS header section is to describe the METS document itself, for example information about the creator of the IP. The requirements for the `metsHdr` element, its sub-elements and attributes are presented in the following table.

ID	Name & Loc	Description & usage	Card & Level
CSIP117	Package header <code>mets/metsHdr</code>	General element for describing the package.	1..1 MUST
CSIP7	Package creation date <code>mets/metsHdr/@CREATEDATE</code>	<code>mets/metsHdr/@CREATEDATE</code> records the date the package was created.	1..1 MUST
CSIP8	Package last modification date <code>mets/metsHdr/@LASTMODDATE</code>	<code>mets/metsHdr/@LASTMODDATE</code> is mandatory when the package has been modified.	0..1 SHOULD
CSIP9	OAIS Package type information <code>mets/metsHdr/@csip:OAISPACKAGETYPE</code>	<code>mets/metsHdr/@csip:OAISPACKAGETYPE</code> is an additional CSIP attribute that declares the type of the IP. See also: OAIS Package type	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP10	Agent <code>mets/metsHdr/agent</code>	A mandatory agent element records the software used to create the package. Other uses of agents may be described in any local implementations that extend the profile.	1..n MUST
CSIP11	Agent role <code>mets/metsHdr/agent[@ROLE='CREATOR']</code>	The mandatory agent element MUST have a <code>@ROLE</code> attribute with the value “CREATOR”.	1..1 MUST
CSIP12	Agent type <code>mets/metsHdr/agent[@TYPE='OTHER']</code>	The mandatory agent element MUST have a <code>@TYPE</code> attribute with the value “OTHER”.	1..1 MUST
CSIP13	Agent other type <code>mets/metsHdr/agent[@OTHERTYPE='SOFTWARE']</code>	The mandatory agent element MUST have a <code>@OTHERTYPE</code> attribute with the value “SOFTWARE”. See also: Other agent type	1..1 MUST
CSIP14	Agent name <code>mets/metsHdr/agent/name</code>	The mandatory agent’s name element records the name of the software tool used to create the IP.	1..1 MUST
CSIP15	Agent additional information <code>mets/metsHdr/agent/note</code>	The mandatory agent’s note element records the version of the tool used to create the IP.	1..1 MUST
CSIP16	Classification of the agent additional information <code>mets/metsHdr/agent/note[@csip:NOTETYPE='SOFTWARE VERSION']</code>	The mandatory agent element’s note child has a <code>@csip:NOTETYPE</code> attribute with a fixed value of “SOFTWARE VERSION”. See also: Note type	1..1 MUST

Example: METS agent example of the mandatory agent

```
<mets:metsHdr CREATEDATE="2018-04-24T14:37:49.602+01:00" LASTMODDATE="
  2018-04-24T14:37:49.602+01:00" RECORDSTATUS="NEW" OAISPACKAGETYPE="SIP"
  >
  <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
    <mets:name>RODA-in</mets:name>
    <mets:note NOTETYPE="SOFTWARE VERSION">2.1.0-beta.7</mets:note>
  </mets:agent>
</mets:metsHdr>
```

5.3.3 Use of the METS descriptive metadata section (element dmdSec)

The purpose of the METS descriptive data section is to embed or refer to files containing descriptive metadata. CSIP is only using referencing of files containing descriptive metadata.

The CSIP as such does not make any assumptions on the use of specific descriptive metadata schemas. As such, implementers are welcome to use descriptive metadata following any standards inside a CSIP package.

Specific elements for which the exact use is fixed within this specification are highlighted in the following table.

ID	Name & Loc	Description & usage	Card & Level
CSIP17	Descriptive metadata <code>mets/dmdSec</code>	Must be used if descriptive metadata for the package content is available. Each descriptive metadata section (<code><dmdSec></code>) contains a single description and must be repeated for multiple descriptions, when available. It is possible to transfer metadata in a package using just the descriptive metadata section and/or administrative metadata section.	0..n SHOULD
CSIP18	Descriptive metadata identifier <code>mets/dmdSec/@ID</code>	An <code>xml:id</code> identifier for the descriptive metadata section (<code><dmdSec></code>) used for internal package references. It must be unique within the package.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP19	Descriptive metadata creation date <code>mets/dmdSec/@CREATED</code>	Creation date of the descriptive metadata in this section.	1..1 MUST
CSIP20	Status of the descriptive metadata <code>mets/dmdSec/@STATUS</code>	Indicates the status of the package using a fixed vocabulary. See also: dmdSec status	0..1 SHOULD
CSIP21	Reference to the document with the descriptive metadata <code>mets/dmdSec/mdRef</code>	Reference to the descriptive metadata file located in the “metadata” section of the IP.	0..1 SHOULD
CSIP22	Type of locator <code>mets/dmdSec/mdRef[@LOCTYPE='URL']</code>	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP23	Type of link <code>mets/dmdSec/mdRef[@xlink:type='simple']</code>	Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP24	Resource location <code>mets/dmdSec/mdRef/@xlink:href</code>	The actual location of the resource. This specification recommends recording a URL type filepath in this attribute.	1..1 MUST
CSIP25	Type of metadata <code>mets/dmdSec/mdRef/@MDTYPE</code>	Specifies the type of metadata in the referenced file. Values are taken from the list provided by the METS.	1..1 MUST
CSIP26	File mime type <code>mets/dmdSec/mdRef/@MIMETYPE</code>	The IANA mime type of the referenced file. See also: IANA media types	1..1 MUST
CSIP27	File size <code>mets/dmdSec/mdRef/@SIZE</code>	Size of the referenced file in bytes.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP28	File creation date <code>mets/dmdSec/ mdRef/@CREATED</code>	The creation date of the referenced file.	1..1 MUST
CSIP29	File checksum <code>mets/dmdSec/ mdRef/@CHECKSUM</code>	The checksum of the referenced file.	1..1 MUST
CSIP30	File checksum type <code>mets/dmdSec/ mdRef/ @CHECKSUMTYPE</code>	The type of checksum following the value list present in the METS-standard which has been used for calculating the checksum for the referenced file.	1..1 MUST

Example: METS example of referencing the descriptive metadata which is described with an EAD document

```
<mets:dmdSec ID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F" CREATED="
2018-04-24T14:37:49.609+01:00">
  <mets:mdRef LOCTYPE="URL" MDTYPE="EAD" type="simple" href="metadata/
descriptive/ead2002.xml" MIMETYPE="application/xml" SIZE="903"
CREATED="2018-04-24T14:37:49.609+01:00" CHECKSUM="
F24263BF09994749F335E1664DCE0086DB6DCA323FDB6996938BCD28EA9E8153"
CHECKSUMTYPE="SHA-256">
  </mets:mdRef>
</mets:dmdSec>
```

5.3.4. Use of the METS administrative metadata section (element amdSec)

METS defines an administrative metadata section, used to embed metadata or provide references to files containing metadata for the information package content. The CSIP encourages the use of references to metadata files, rather than embedding metadata in the `amdSec` element, and the examples reflect this.

The administrative metadata section contains four sub-sections each used to record different types of metadata for package content: - technical metadata (element `techMD`) records technical metadata; - rights metadata (element `rightsMD`) records intellectual property rights information; - source metadata (element `sourceMD`) records descriptive, technical or rights metadata for an analog source for a digital

library object; and - digital provenance metadata (element `digiprovMD`) records digital preservation information, e.g. audit information covering a digital library object's life-cycle.

The CSIP only describes use of the elements `digiprovMD` and `rightsMD`. The population of the other metadata sections are left to local policy and practise.

The CSIP (and METS) categorises preservation metadata as administrative metadata, specifically Digital Provenance metadata (following the available guidelines published by the PREMIS EC guidelines: <http://www.loc.gov/standards/premis/guidelines2017-premismets.pdf>), hence all preservation metadata should be referenced from a `digiprovMD` element within the `amdSec`.

The METS `amdSec` element SHOULD include references to all relevant metadata located in the folder "metadata/preservation". The package level `METS.xml` file SHOULD only reference package level preservation metadata. Representation level `METS.xml` files SHOULD only reference representation level preservation metadata.

The specific requirements for the `amdSec` element, its sub-elements and attributes are presented in the following table.

ID	Name & Loc	Description & usage	Card & Level
CSIP31	Administrative metadata <code>mets/amdSec</code>	If administrative / preservation metadata is available, it must be described using the administrative metadata section (<code><amdSec></code>) element. All administrative metadata is present in a single <code><amdSec></code> element. It is possible to transfer metadata in a package using just the descriptive metadata section and/or administrative metadata section.	0..1 SHOULD
CSIP32	Digital provenance metadata <code>mets/amdSec/digiprovMD</code>	For recording information about preservation the standard PREMIS is used. It is mandatory to include one <code><digiprovMD></code> element for each piece of PREMIS metadata. The use if PREMIS in METS is following the recommendations in 2017 version of PREMIS in METS Guidelines	0..n SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP33	Digital provenance metadata identifier <code>mets/amdSec/digiprovMD/@ID</code>	An <code>xml:id</code> identifier for the digital provenance metadata section <code>mets/amdSec/digiprovMD</code> used for internal package references. It must be unique within the package.	1..1 MUST
CSIP34	Status of the digital provenance metadata <code>mets/amdSec/digiprovMD/@STATUS</code>	Indicates the status of the package using a fixed vocabulary. See also: dmdSec status	0..1 SHOULD
CSIP35	Reference to the document with the digital provenance metadata <code>mets/amdSec/digiprovMD/mdRef</code>	Reference to the digital provenance metadata file stored in the “metadata” section of the IP.	0..1 SHOULD
CSIP36	Type of locator <code>mets/amdSec/digiprovMD/mdRef[@LOCTYPE='URL']</code>	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP37	Type of link <code>mets/amdSec/digiprovMD/mdRef[@xlink:type='simple']</code>	Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP38	Resource location <code>mets/amdSec/digiprovMD/mdRef/@xlink:href</code>	The actual location of the resource. This specification recommends recording a URL type filepath within this attribute.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP39	Type of metadata mets/amdSec/ digiprovMD/ mdRef/@MDTYPE	Specifies the type of metadata in the referenced file. Values are taken from the list provided by the METS.	1..1 MUST
CSIP40	File mime type mets/amdSec/ digiprovMD/ mdRef/@MIMETYPE	The IANA mime type for the referenced file. See also: IANA media types	1..1 MUST
CSIP41	File size mets/amdSec/ digiprovMD/ mdRef/@SIZE	Size of the referenced file in bytes.	1..1 MUST
CSIP42	File creation date mets/amdSec/ digiprovMD/ mdRef/@CREATED	Creation date of the referenced file.	1..1 MUST
CSIP43	File checksum mets/amdSec/ digiprovMD/ mdRef/@CHECKSUM	The checksum of the referenced file.	1..1 MUST
CSIP44	File checksum type mets/amdSec/ digiprovMD/ mdRef/ @CHECKSUMTYPE	The type of checksum following the value list present in the METS-standard which has been used for calculating the checksum for the referenced file.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP45	Rights metadata <code>mets/amdSec/</code> <code>rightsMD</code>	A simple rights statement may be used to describe general permissions for the package. Individual representations should state their specific rights in their representation METS file. Available standards include RightsStatements.org , Europeana rights statements info), METS Rights Schema created and maintained by the METS Board, the rights part of PREMIS as well as own local rights statements in use.	0..n MAY
CSIP46	Rights metadata identifier <code>mets/amdSec/</code> <code>rightsMD/@ID</code>	An <code>xml:id</code> identifier for the rights metadata section (<code><rightsMD></code>) used for internal package references. It must be unique within the package.	1..1 MUST
CSIP47	Status of the rights metadata <code>mets/</code> <code>amdSec/rightsMD</code> <code>/@STATUS</code>	Indicates the status of the package using a fixed vocabulary. See also: dmdSec status	0..1 SHOULD
CSIP48	Reference to the document with the rights metadata <code>mets/amdSec/</code> <code>rightsMD/mdRef</code>	Reference to the rights metadata file stored in the “metadata” section of the IP.	0..1 SHOULD
CSIP49	Type of locator <code>mets/amdSec/</code> <code>rightsMD/mdRef[</code> <code>@LOCTYPE=URL]</code>	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP50	<code>mets/amdSec/</code> <code>rightsMD/mdRef[</code> <code>@xlink:type='</code> <code>simple']</code>	Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP51	Resource location <code>mets/amdSec/</code> <code>rightsMD/mdRef/</code> <code>@xlink:href</code>	The actual location of the resource. We recommend recording a URL type filepath within this attribute.	1..1 MUST
CSIP52	Type of metadata <code>mets/amdSec/</code> <code>rightsMD/mdRef/</code> <code>@MDTYPE</code>	Specifies the type of metadata in the referenced file. Value is taken from the list provided by the METS.	1..1 MUST
CSIP53	File mime type <code>mets/amdSec/</code> <code>rightsMD/mdRef/</code> <code>@MIMETYPE</code>	The IANA mime type for the referenced file. See also: IANA media types	1..1 MUST
CSIP54	File size <code>mets/</code> <code>amdSec/rightsMD</code> <code>/mdRef/@SIZE</code>	Size of the referenced file in bytes.	1..1 MUST
CSIP55	File creation date <code>mets/amdSec/</code> <code>rightsMD/mdRef/</code> <code>@CREATED</code>	Creation date of the referenced file.	1..1 MUST
CSIP56	File checksum <code>mets/amdSec/</code> <code>rightsMD/mdRef/</code> <code>@CHECKSUM</code>	The checksum of the referenced file.	1..1 MUST
CSIP57	File checksum type <code>mets/amdSec/</code> <code>rightsMD/mdRef/</code> <code>@CHECKSUMTYPE</code>	The type of checksum following the value list present in the METS-standard which has been used for calculating the checksum for the referenced file.	1..1 MUST

Example: METS example of referencing preservation metadata in the form of PREMIS metadata for describing the preservation objects and the events pertaining to the objects

```
<mets:amdSec>
  <mets:digiprovMD ID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F" CREATED=
    "2018-04-24T14:37:52.783+01:00">
```



```
<mets:mdRef LOCTYPE="URL" type="simple" href="metadata/preservation/
  premis1.xml" MDTYPE="PREMIS" MDTYPEVERSION="3.0" MIMETYPE="text/xml
  " SIZE="1211" CREATED="2018-04-24T14:37:52.783+01:00" CHECKSUM="8
  aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
  CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
</mets:mdRef>
</mets:digiprovMD>
<mets:digiprovMD ID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3" CREATED=
  "2018-04-24T14:47:52.783+01:00">
  <mets:mdRef LOCTYPE="URL" type="simple" href="metadata/preservation/
    premis2.xml" MDTYPE="PREMIS" MDTYPEVERSION="3.0" MIMETYPE="text/xml
    " SIZE="2854" CREATED="2018-04-24T14:37:52.783+01:00" CHECKSUM="
    d1dfa585dcc9d87268069dc58d5e47956434ec3db4087a75a3885d287f15126f"
    CHECKSUMTYPE="SHA-256" LABEL="premis2.xml">
  </mets:mdRef>
</mets:digiprovMD>
</mets:amdSec>
```

5.3.5. Use of the METS file section (element fileSec)

The CSIP does not make the use of the METS fileSec element mandatory, but it is strongly recommended. The fileSec should describe every component of the IP not already described in the amdSec and dmdSec elements. Location and checksum values must be provided for all file entries.

The METS file section serves as a manifest, allowing users to ensure all files are present and that a package is complete, as well as testing the integrity of package files using checksum values.

The files are described in a file group element named `fileGrp`. The `fileGrp` element contains the descriptions of the files including their checksum and location. It is possible to nest `fileGrp` elements within other `fileGrp` elements, however the CSIP states that hierarchical nesting SHOULD NOT be used.

The CSIP requires that both package and representation level METS files include at least three file group (`fileGrp`) elements. The CSIP divides the referenced files into categories, i.e. documentation, schemas, data, each contained within its own `fileGrp` element. It is possible to add own additional file groups following the same file groups requirements described in CSIP. Representation level METS files should not reference files outside of their representation. That is they should not contain references to package level files or files from other representations. The package level METS file should contain a `fileGrp` element for each representation which contains a single reference to the representation's METS.xml file. This is to ensure that the package lists all representations and their locations for completeness and integrity checks.

The specific requirements for elements, sub-elements and attributes are listed in the following table. Note that the use of the stream and transformFile elements is not discussed below. Implementers wishing to use either of these METS elements should follow the requirements in the METS documentation.

ID	Name & Loc	Description & usage	Card & Level
CSIP58	File section <code>mets/fileSec</code>	The transferred content is placed in the file section in different file group elements, described in other requirements. Only a single file section (<fileSec>) element should be present. It is possible to transfer just descriptive metadata and/or administrative metadata without files placed in this section.	0..1 SHOULD
CSIP59	File section identifier <code>mets/fileSec/@ID</code>	An <code>xml:id</code> identifier for the file section used for internal package references. It must be unique within the package.	1..1 MUST
CSIP60	Documentation file group <code>mets/fileSec/fileGrp[@USE='Documentation']</code>	All documentation pertaining to the transferred content is placed in one or more file group elements with <code>mets/fileSec/fileGrp/@USE</code> attribute value “Documentation”. See also: File group names	1..n MUST
CSIP113	Schema file group <code>mets/fileSec/fileGrp[@USE='Schemas']</code>	All XML schemas used in the information package should be referenced from one or more file groups with <code>mets/fileSec/fileGrp/@USE</code> attribute value “Schemas”. See also: File group names	1..n MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP114	Representations file group <code> mets/fileSec/fileGrp[@USE='Representations']</code>	A pointer to the METS document describing the representation or pointers to the content being transferred must be present in one or more file groups with <code> mets/fileSec/fileGrp/@USE</code> attribute value “Representations”. See also: File group names	1..n MUST
CSIP61	Reference to administrative metadata <code> mets/fileSec/fileGrp/@ADMID</code>	If administrative metadata has been provided at file group <code> mets/fileSec/fileGrp</code> level this attribute refers to its administrative metadata section by ID.	0..1 MAY
CSIP62	Content Information Type Specification <code> mets/fileSec/fileGrp[@LABEL='Representations']/@csip:CONTENTINFORMATI</code>	An added attribute which states the name of the content information type specification used to create the package. The vocabulary will evolve under the curation of the DILCIS Board as additional content information type specifications are developed. This attribute is mandatory when the <code> mets/fileSec/fileGrp/@LABEL</code> attribute value is “Representations”. When the “Package type” value is “Mixed” and/or the file group describes a “Representation”, then this element states the content information type specification used for the file group. See also: Content information type specification	0..1 SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP63	Other Content Information Type Specification <code>mets/fileSec/fileGrp[@csip:CONTENTINFORMATIONTYPE= 'OTHER']/@csip:OTHERCONTENTINFORMATIONTYPE</code>	When the <code>mets/fileSec/fileGrp[@csip:CONTENTINFORMATIONTYPE= 'OTHER']</code> attribute has the value “OTHER” the <code>mets/fileSec/fileGrp[@csip:OTHERCONTENTINFORMATIONTYPE</code> must state a value for the Content Information Type Specification used.	0..1 MAY
CSIP64	Description of the use of the file group <code>mets/fileSec/fileGrp/@USE</code>	The value in the <code>mets/fileSec/fileGrp/@USE</code> is the name of the whole folder structure to the data, e.g “Documentation”, “Schemas”, “Representations/preingest” or “Representations/submission/data”.	1..1 MUST
CSIP65	File group identifier <code>mets/fileSec/fileGrp/@ID</code>	An <code>xml:id</code> identifier for the file group used for internal package references. It must be unique within the package.	1..1 MUST
CSIP66	File <code>mets/fileSec/fileGrp/file</code>	The file group (<code><fileGrp></code>) contains the file elements which describe the file objects.	1..n MUST
CSIP67	File identifier <code>mets/fileSec/fileGrp/file/@ID</code>	A unique <code>xml:id</code> identifier for this file across the package.	1..1 MUST
CSIP68	File mimetype <code>mets/fileSec/fileGrp/file/@MIMETYPE</code>	The IANA mime type for the referenced file. See also: IANA media types	1..1 MUST
CSIP69	File size <code>mets/fileSec/fileGrp/file/@SIZE</code>	Size of the referenced file in bytes.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP70	File creation date <code>mets/fileSec/ fileGrp/file/ @CREATED</code>	Creation date of the referenced file.	1..1 MUST
CSIP71	File checksum <code>mets/ fileSec/fileGrp/ file/@CHECKSUM</code>	The checksum of the referenced file.	1..1 MUST
CSIP72	File checksum type <code>mets/fileSec/ fileGrp/file/ @CHECKSUMTYPE</code>	The type of checksum following the value list present in the METS-standard which has been used for calculating the checksum for the referenced file.	1..1 MUST
CSIP73	File original identification <code>mets/ fileSec/fileGrp/ file/@OWNERID</code>	If an identifier for the file was supplied by the owner it can be recorded in this attribute.	0..1 MAY
CSIP74	File reference to administrative metadata <code>mets/ fileSec/fileGrp/ file/@ADMID</code>	If administrative metadata has been provided for the file this attribute refers to the file's administrative metadata by ID.	0..1 MAY
CSIP75	File reference to descriptive metadata <code>mets/ fileSec/fileGrp/ file/@DMDID</code>	If descriptive metadata has been provided per file this attribute refers to the file's descriptive metadata by ID.	0..1 MAY
CSIP76	File locator reference <code>mets/ fileSec/fileGrp/ file/FLocat</code>	The location of each external file must be defined by the file location <code><FLocat></code> element using the same rules as for referencing metadata files. All references to files should be made using the XLink href attribute and the file protocol using the relative location of the file.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP77	Type of locator <code>mets/fileSec/ fileGrp/file/ FLocat[@LOCTYPE ='URL']</code>	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP78	Type of link <code>mets/fileSec/ fileGrp/file/ FLocat[@xlink: type='simple']</code>	Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP79	Resource location <code>mets/fileSec/ fileGrp/file/ FLocat/@xlink: href</code>	The actual location of the resource. We recommend recording a URL type filepath within this attribute.	1..1 MUST

Example: METS example of how the structuring of the data is made in the file section

```
<mets:fileSec ID="uuid-CA580D47-8C8B-4E91-ABD5-142EBBE15B84">
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H" USE="
    Documentation">
    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A81" MIMETYPE="
      application/vnd.openxmlformats-officedocument.wordprocessingml.
      document" SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
      CHECKSUM="91
      B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
      CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="Documentation/File.
        docx">
      </mets:FLocat>
    </mets:file>
    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A82" MIMETYPE="
      application/vnd.openxmlformats-officedocument.wordprocessingml.
      document" SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
      CHECKSUM="91
      B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
      CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="Documentation/File2.
```

```

        docx">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F" USE="
    Schemas">
    <mets:file ID="uuid-A1B7B0DA-E129-48EF-B431-E553F2977FD6" MIMETYPE="
      text/xsd" SIZE="123917" CREATED="2018-04-24T14:37:49.617+01:00"
      CHECKSUM="0
      BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
      CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="schemas/ead2002.xsd">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86G" USE="
    Representations/Submission/Data" CONTENTINFORMATIONTYPE="SIARDDK">
    <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FD" MIMETYPE="
      xml" SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
      CHECKSUM="7176
      A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
      CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-
      F941CBB138B3 uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
      <mets:FLocat LOCTYPE="URL" type="simple" href="representations/
        Submission/Data/SIARD.xml">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
</mets:fileSec>

```

Example: METS example of how the structuring of the data including representations are made in the file section

```

<mets:fileSec ID="uuid-CA580D47-8C8B-4E91-ABD5-142EBBE15B84">
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H" USE="
    Documentation">
    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A81" MIMETYPE="
      application/vnd.openxmlformats-officedocument.wordprocessingml.
      document" SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
      CHECKSUM="91
      B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
      CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="documentation/File.

```

```

        docx">
      </mets:FLocat>
    </mets:file>
    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A82" MIMETYPE="
      application/vnd.openxmlformats-officedocument.wordprocessingml.
      document" SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
      CHECKSUM="91
      B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
      CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="documentation/File2.
        docx">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F" USE="
    schemas">
    <mets:file ID="uuid-A1B7B0DA-E129-48EF-B431-E553F2977FD6" MIMETYPE="
      text/xsd" SIZE="123917" CREATED="2018-04-24T14:37:49.617+01:00"
      CHECKSUM="0
      BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
      CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="schemas/ead2002.xsd">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C277" USE="
    Representations/preingest" CONTENTINFORMATIONTYPE="OTHER"
    OTHERCONTENTINFORMATIONTYPE="Access database">
    <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FE" MIMETYPE="
      xml" SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
      CHECKSUM="7176
      A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
      CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-
      F941CBB138B3 uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
      <mets:FLocat LOCTYPE="URL" type="simple" href="representations/
        preingest/METS.xml">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C278" USE="
    Representations/submission/data" CONTENTINFORMATIONTYPE="SIARDDK"
    ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F uuid-48C18DD8
    -2561-4315-AC39-F941CBB138B3">

```



```

<mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FF" MIMETYPE="
  application/xml" SIZE="1338744" CREATED="2018-04-24T14
  :37:49.617+01:00" CHECKSUM="7176
  A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
  CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-
  F941CBB138B3 uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
  <mets:FLocat LOCTYPE="URL" type="simple" href="representations/
    Submission/METS.xml">
  </mets:FLocat>
</mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C279" USE="
  Representations/ingest/data" CONTENTINFORMATIONTYPE="SIARD1" ADMID="
  uuid-9124DA4D-3736-4F69-8355-EB79A22E943G uuid-48C18DD8-2561-4315-
  AC39-F941CBB138B4">
  <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FG" MIMETYPE="
    application/xml" SIZE="1338744" CREATED="2018-04-24T14
    :37:49.617+01:00" CHECKSUM="7176
    A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
    CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-
    F941CBB138B3 uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
    <mets:FLocat LOCTYPE="URL" type="simple" href="representations/
      ingest/METS.xml">
    </mets:FLocat>
  </mets:file>
</mets:fileGrp>
</mets:fileSec>

```

5.3.6. Use of the METS structural map (<structMap>)

The METS structural map element is the only mandatory element in the METS specification. It provides an overview of the components described in the METS document. It can also link the elements in the structure to associated content files and metadata. In the CSIP the <structMap> describes the higher level structure of all the content in the package and may link to representations.

The CSIP requires the inclusion of one mandatory structural map according to the principles described below. However, implementers are welcome to define additional structural maps for their internal purposes by repeating the <structMap> element. The most crucial requirements for the CSIP mandated structural map are as follows:

- The <structMap> element's @LABEL attribute is mandatory and must have the value "CSIP". The @LABEL attribute distinguishes the CSIP mandated structural maps from other structural

maps. NOTE this means that the “CSIP” @LABEL value should be treated as a unique id and not applied to other <structMap> elements;

- The package structure is documented using METS division <div> elements which can be arranged in nested tree structures. The CSIP organises it’s constituent <div> elements in a single root structural <div> element.
 - This top level structural <div> element must use the package identifier as the value of its @LABEL attribute.
- The internal structure of the structural map (expressed by <div> elements) follows the CSIP high level physical structure described in Section 4, and groups metadata, representations, schemas, documentation and user-defined folders into discrete <div> elements;
 - All <div> elements must have an @LABEL attribute with the value identical to the folder name, e.g. “metadata”;
- When both package and representation METS files are present, the structural map in the Package METS document:
 - References the <fileGrp> which describes every file in every folder, with the exception of the content of the representation folders
 - Lists all representations (as separate <div> elements)
 - Lists only the appropriate representation METS file using the <mptr> element as the content of the representation
- The structural map in the representations METS file uses the structural map in an identical manner, describing all representation structural map with no exceptions

The specific requirements for elements, sub-elements and attributes are listed in the following table. Note that the <area>, <seq> and <par> elements are not discussed below.

ID	Name & Loc	Description & usage	Card & Level
CSIP80	Structural description of the package mets/structMap	The structural map <structMap> element is the only mandatory element in the METS. The <structMap> in the CSIP describes the highest logical structure of the IP. Each METS file must include ONE structural map <structMap> element used exactly as described here. Institutions can add their own additional custom structural maps as separate <structMap> sections.	1..n MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP81	Type of structural description <code>mets/structMap[@TYPE='PHYSICAL']</code>	The <code>mets/structMap/@TYPE</code> attribute must take the value “PHYSICAL” from the vocabulary. See also: Structural map typing	1..1 MUST
CSIP82	Name of the structural description <code>mets/structMap[@LABEL='CSIP']</code>	The <code>mets/structMap/@LABEL</code> attribute value is set to “CSIP” from the vocabulary. See also: Structural map label	1..1 MUST
CSIP83	Structural description identifier <code>mets/structMap[@LABEL='CSIP']/@ID</code>	An <code>xml:id</code> identifier for the structural description (structMap) used for internal package references. It must be unique within the package.	1..1 MUST
CSIP84	Main structural division <code>mets/structMap[@LABEL='CSIP']/div</code>	The structural map comprises a single division.	1..1 MUST
CSIP85	Main structural division identifier <code>mets/structMap[@LABEL='CSIP']/div/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP86	Main structural division label <code>mets/structMap[@LABEL='CSIP']/div/@LABEL</code>	The package’s top-level structural division <code><div></code> element’s <code>@LABEL</code> attribute value must be identical to the package identifier, i.e. the same value as the <code>mets/@OBJID</code> attribute.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP88	Metadata division <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Metadata']</code>	The metadata referenced in the administrative and/or descriptive metadata section is described in the structural map with one sub division. When the transfer consist of only administrative and/or descriptive metadata this is the only sub division that occurs.	1..1 MUST
CSIP89	Metadata division identifier <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Metadata']/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP90	Metadata division label <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Metadata']</code>	The metadata division <code><div></code> element's <code>@LABEL</code> attribute value must be "Metadata". See also: File group names	1..1 MUST
CSIP91	Metadata division administrative metadata referencing <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Metadata']/ @ADMID</code>	When there is administrative metadata and the <code>amdSec</code> is present, all administrative metadata MUST be referenced via the administrative sections different identifiers. All of the <code><amdSec></code> identifiers are listed in a single <code>@AMDID</code> using spaces as delimiters.	0..1 SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP92	Metadata division descriptive metadata referencing <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Metadata']/ @DMDID</code>	When there are descriptive metadata and one or more dmdSec is present, all descriptive metadata MUST be referenced via the descriptive section identifiers. Every <dmdSec> identifier is listed in a single @DMDID attribute using spaces as delimiters.	0..1 SHOULD
CSIP93	Documentation division <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Documentation']</code>	The documentation referenced in the file section file groups is described in the structural map with one sub division.	0..1 SHOULD
CSIP94	Documentation division identifier <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Documentation']/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP95	Documentation division label <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Documentation']</code>	The documentation division <div> element in the package uses the value “Documentation” from the vocabulary as the value for the @LABEL attribute. See also: File group names	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP96	Documentation file referencing <code>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']/fptr</code>	All file groups containing documentation described in the package are referenced via the relevant file group identifiers. There MUST be one file group reference per <code><fptr></code> element.	0..n MUST
CSIP116	Documentation file group reference pointer <code>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']/fptr/@FILEID</code>	A reference, by ID, to the “Documentation” file group. Related to the requirements CSIP60 which describes the “Documentation” file group and CSIP65 which describes the file group identifier.	1..1 MUST
CSIP97	Schema division <code>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']</code>	The schemas referenced in the file section file groups are described in the structural map within a single sub-division.	0..1 SHOULD
CSIP98	Schema division identifier <code>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP99	Schema division label <code>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']</code>	The schema division <code><div></code> element’s <code>@LABEL</code> attribute has the value “Schemas” from the vocabulary. See also: File group names	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP100	Schema file reference <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Schemas']/fptr</code>	All file groups containing schemas described in the package are referenced via the relevant file group identifiers. One file group reference per fptr-element	0..n MUST
CSIP118	Schema file group reference <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Schemas']/fptr /@FILEID</code>	The pointer to the identifier for the “Schema” file group. Related to the requirements CSIP113 which describes the “Schema” file group and CSIP65 which describes the file group identifier.	1..1 MUST
CSIP101	Content division <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= ' Representations ']</code>	When no representations are present the content referenced in the file section file group with @USE attribute value “Representations” is described in the structural map as a single sub division.	0..1 SHOULD
CSIP102	Content division identifier <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= ' Representations ']/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP103	Content division label <code>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']</code>	The package's content division <code><div></code> element must have the <code>@LABEL</code> attribute value "Representations", taken from the vocabulary. See also: File group names	1..1 MUST
CSIP104	Content division file references <code>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']/fptr</code>	All file groups containing content described in the package are referenced via the relevant file group identifiers. One file group reference per fptr-element	0..n MUST
CSIP119	Content division file group references <code>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']/fptr/@FILEID</code>	The pointer to the identifier for the "Representations" file group. Related to the requirements CSIP114 which describes the "Representations" file group and CSIP65 which describes the file group identifier.	1..1 MUST
CSIP105	Representation division <code>mets/structMap[@LABEL='CSIP']/div/div</code>	When a package consists of multiple representations, each described by a representation level METS.xml document, there is a discrete representation div element for each representation. Each representation div references the representation level METS.xml document, documenting the structure of the package and its constituent representations.	0..n SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP106	Representations division identifier <code>mets/structMap[@LABEL='CSIP']/ div/div/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP107	Representations division label <code>mets/structMap[@LABEL='CSIP']/ div/div/@LABEL</code>	The package's representation division <code><div></code> element <code>@LABEL</code> attribute value must be the path to the representation level METS document. This requirement gives the same value to be used as the requirement named "File group identifier" (CSIP64) See also: File group names	1..1 MUST
CSIP108	Representations division file referencing <code>mets/structMap[@LABEL='CSIP']/ div/div/mptr/ @xlink:title</code>	The file group containing the files described in the package are referenced via the relevant file group identifier. Related to the requirements CSIP114 which describes the "Representations" file group and CSIP65 which describes the file group identifier.	1..1 MUST
CSIP109	Representation METS pointer <code>mets/structMap[@LABEL='CSIP']/ div/div/mptr</code>	The division <code><div></code> of the specific representation includes one occurrence of the METS pointer <code><mptr></code> element, pointing to the appropriate representation METS file.	1..1 MUST
CSIP110	Resource location <code>mets/structMap/ div/div/mptr/ @xlink:href</code>	The actual location of the resource. We recommend recording a URL type filepath within this attribute.	1..1 MUST
CSIP111	Type of link <code>mets/ structMap/div/ div/mptr[@xlink: :type='simple']</code>	Attribute used with the value "simple". Value list is maintained by the xlink standard.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP112	Type of locator mets/structMap/ div/div/mptr[@LOCTYPE='URL']	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST

Example: METS example of the mandatory structural map

```
<mets:structMap ID="uuid-1465D250-0A24-4714-9555-5C1211722FB8" TYPE="
  PHYSICAL" LABEL="CSIP">
  <mets:div ID="uuid-638362BC-65D9-4DA7-9457-5156B3965A18" LABEL="uuid
    -4422c185-5407-4918-83b1-7abfa77de182">
    <mets:div ID="uuid-A4E1C5B6-CD9B-43EF-8F0C-3FD3AB688F81" LABEL="
      Metadata" ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F uuid-48
      C18DD8-2561-4315-AC39-F941CBB138B3" DMDID="uuid-906F4F12-BA52-4779-
      AE2C-178F9206111F">
      </mets:div>
      <mets:div ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86I" LABEL="
        Documentation">
        <mets:fptr FILEID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H">
        </mets:fptr>
        </mets:div>
        <mets:div ID="uuid-26757DC2-4C0F-4431-85B5-5943D1AB5CA3" LABEL="
          Schemas">
          <mets:fptr FILEID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F">
          </mets:fptr>
          </mets:div>
          <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6736" LABEL="
            Representations">
            <mets:fptr FILEID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86G">
            </mets:fptr>
            </mets:div>
          </mets:div>
        </mets:div>
      </mets:structMap>
```

Example: METS example of the mandatory structural map including representations

```
<mets:structMap ID="uuid-1465D250-0A24-4714-9555-5C1211722FB8" TYPE="
  PHYSICAL" LABEL="CSIP">
  <mets:div ID="uuid-638362BC-65D9-4DA7-9457-5156B3965A18" LABEL="uuid
    -4422c185-5407-4918-83b1-7abfa77de182">
```

```

<mets:div ID="uuid-A4E1C5B6-CD9B-43EF-8F0C-3FD3AB688F81" LABEL="
  Metadata" ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F uuid-48
  C18DD8-2561-4315-AC39-F941CBB138B3 uuid-9124DA4D-3736-4F69-8355-
  EB79A22E943G uuid-48C18DD8-2561-4315-AC39-F941CBB138B4" DMDID="uuid
  -906F4F12-BA52-4779-AE2C-178F9206111F">
</mets:div>
<mets:div ID="uuid-26757DC2-4C0F-4431-85B5-5943D1AB5CA3" LABEL="
  Schemas">
  <mets:fptr FILEID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F">
  </mets:fptr>
</mets:div>
<mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6737" LABEL="
  representations/preingest">
  <mets:mptr LOCTYPE="URL" type="simple" href="representations/
  preingest/METS.xml" title="uuid-5811D494-6045-4741-924C-
  A1CFA340C277">
  </mets:mptr>
</mets:div>
<mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6736" LABEL="
  representations/submission">
  <mets:mptr LOCTYPE="URL" type="simple" href="representations/
  submission/METS.xml" title="uuid-5811D494-6045-4741-924C-
  A1CFA340C278">
  </mets:mptr>
</mets:div>
<mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6738" LABEL="
  representations/ingest">
  <mets:mptr LOCTYPE="URL" type="simple" href="representations/ingest/
  METS.xml" title="uuid-5811D494-6045-4741-924C-A1CFA340C279">
  </mets:mptr>
</mets:div>
</mets:div>
</mets:structMap>

```

5.4. Use of PREMIS

The CSIP recommends and advocates the use of the PREservation Metadata Implementation Strategies (PREMIS, information available at <http://www.loc.gov/standards/premis/>) metadata standard for recording preservation and technical metadata about digital objects contained within CSIP Information Packages. The CSIP implements version 3.0 of the PREMIS Data Dictionary. Note that use of PREMIS is not mandatory.

We strongly recommend keeping PREMIS metadata in discrete PREMIS XML files inside the IP. The PREMIS metadata can be included in the IP in separate files, and there is no convention regarding the naming and numbering of the PREMIS files. Implementations can choose to either store all preservation metadata in a single PREMIS file or split them into multiple files. The only requirement in this case is that all PREMIS files must be listed in the appropriate METS file, i.e. package PREMIS files from the package METS file and representation PREMIS files from the representation METS files, and referenced in the METS file(s) using the mdRef attributes and elements.

Therefore, the main recommendation of the CSIP is that preservation metadata are included in the information package in PREMIS format. Although this is not mandatory, all tools claiming to be able to validate CSIP compliant Information Packages must also be able to validate PREMIS metadata once it exists within the package. The two high level requirements for use of PREMIS in Common Specification IPs are that:

- All preservation metadata is created according to official PREMIS guidelines;
- All PREMIS metadata is referenced from the amdSec/digiprovMD element of the appropriate METS file.

Further, to enhance the interoperability scope of the CSIP and to strengthen management of IPs in an archive, this specification imposes additional requirements with regard to use of PREMIS for describing Information Packages. The principles adopted in the CSIP for deciding the additional PREMIS semantic units required are:

- PREMIS should be used to record detailed technical metadata;
- Technical information should be included in PREMIS metadata by using the possibility of extension schemas in PREMIS;
- Information about agents carrying out preservation actions must be recorded in the PREMIS metadata. (The METS agents describe agents who are relevant for generic IP level events, such as the creation or submission of the package, for example);
- Event descriptions should be included in PREMIS metadata. Use of the official PREMIS event vocabulary (<http://id.loc.gov/vocabulary/preservation/eventType.html>) is recommended;
- Detailed rights information should be included in PREMIS. High level rights information in METS indicates restrictions. Detailed, object-specific, rights information will be included in the PREMIS metadata;
- File format information for all files should be included as PUID values in the appropriate PREMIS semantic units.

Vocabularies

This specification does not present a definitive list of vocabularies for use with PREMIS semantic units but does recommend the use of the Library of Congress vocabularies developed specifically to provide values for various PREMIS semantic units. All relevant vocabularies is presented in the PREMIS Data Dictionary.

Identifiers

In PREMIS each of the entities (objects, events, agents, rights) are identified by a generic set of identifier containers. These containers follow an identical syntax and structure consisting of an [entity]Identifier container holding two semantic units:

- [entity]IdentifierType
- [entity]IdentifierValue

The PREMIS data dictionary recognizes that the use of identifier types is an implementation specific issue and does not recommend or require particular vocabularies for identifier types. The Library of Congress has a Standard Identifiers Scheme: <http://id.loc.gov/vocabulary/identifiers.html>, and the CSIP recommends its use in lieu of implementation specific vocabularies.

6. Implementation Considerations

This Section touches on some additional issues which are relevant in respect to implementing the CSIP in real-life scenarios.

6.1 Content Information Type Specifications

6.1.1 What is a Content Information Type Specification?

A Content Information Type Specification is a mechanism used to extend the scope of the CSIP by defining additional requirements for specific Content Information Types. The OAIS Reference Model defines Content Information as “A set of information that is the original target of preservation or that includes part or all of that information. It is an Information Object composed of its Content Data Object and its Representation Information”.

Content Information Types can be regarded as categories of Content Information, e.g. relational databases, scientific data or digitised maps. A Content Information Type Specification defines the

format and structure, mainly in regard to the Information Object, within an Information Package. This facilitates interoperability when exchanging specific Content Information Types.

The following elements should be at the core of a Content Information Type Specification:

- Required file formats for content data;
- Specification of information package structure, i.e. specific sub-folders under the “Data” folder for particular categories of content data;
- Requirements for specific representation metadata that should be available in PREMIS for understanding and rendering the Content Data Object;
- Details of specific (binary) documentation or other components (e.g. software, emulators, etc.) which must be available for rendering and understanding the Content Data Object.

Pragmatically it may not be sufficient to deal only with the Information Object. For complex Content Information Types or large IPs it may be necessary to provide explicit requirements for other metadata (descriptive, administrative) relevant to the specific content type. For example, the ERMS Content Information Type Specification does specify a method for referencing data (i.e. computer files) from descriptive metadata in ERMS format, to ensure the integrity of data and metadata. Stating these requirements in a general specification allows archival institutions receiving SIPs including ERMS extracts or whole systems to understand and validate potentially complex information packages.

Content Information Type Specification can be domain specific and there might be multiple specifications to cover a particular domain. For example, archival institutions might define a Content Information Type Specification for archiving web sites with descriptive metadata in EAD format, while libraries might define a specification for archiving web sites with MARC metadata.

6.1.2 Maintaining Content Information Type Specifications

We hope to see the development of many Content Information Type Specifications with the wider community to creating new specifications for domains of interest to them. The DILCIS Board aims to work with the community to maintain a list of available specifications following some simple principles:

- The DILCIS Board is responsible for establishing reasonable guidelines and quality requirements for new Content Information Type Specifications, and publishing these on the Board website;
- The Board has the responsibility and mandate to manage a registry of available Content Information Type Specifications which meet the guidelines and quality requirements;
- The Board does NOT take ownership of or have responsibility for maintaining and sustaining any Content Information Type Specifications;
- There shall be no limitations to who is allowed to propose additional Content Information Type Specifications; and

- To ensure the quality of available specifications, the Board validates each proposed specification against the guidelines and quality requirements mentioned above. The validation shall be carried out free of charge and within a reasonable time-frame.

6.2. Handling large packages

By default a Common Specification IP should be contained in a single folder or archive file. However, the amount of data and metadata within a single IP can easily grow to several GB or even TB and become difficult to manage and inefficient to process e.g. due to insufficient storage capacity.

The Common Specification can, in principle, be extended in different ways to support the segmentation of large packages into more manageable parts.

The handling of large packages will be specified in a later version of CSIP. Please be aware that this will be a “recommended approach” and, at this point in time, it is not a part of the CSIP instead segmentation of large packages is depending on the local implementation. It is also not expected that all tools will support segmentation of large packages.

6.3 Handling descriptive metadata within the Common Specification

Descriptive metadata are used to describe the intellectual contents of archival holdings, and they support finding and understanding individual information packages. The CSIP allows essentially for the inclusion of any kind of descriptive metadata in the IP. However, it is required that all descriptive metadata must be placed into the “metadata” folder of the IP, and that it is recommended (should) to also exploit the possibility of creating a specific sub-folder “descriptive” as seen in Figure 11 below (cf. EAD.xml).

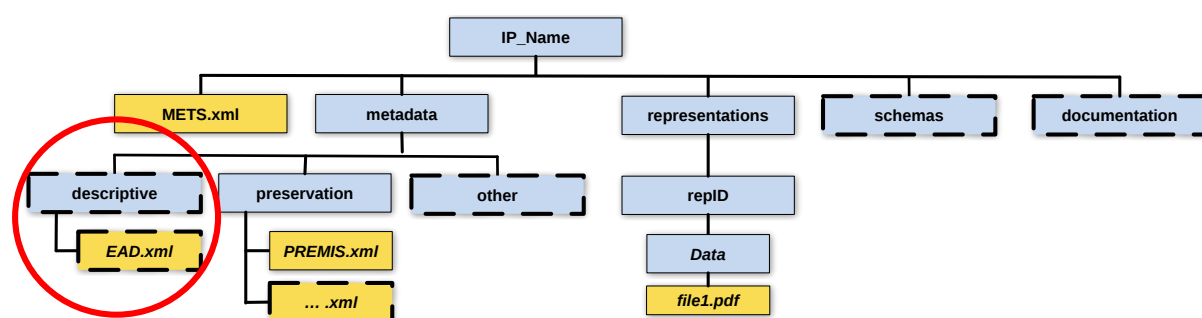


Figure 11: E-ARK IP descriptive metadata

Further, all descriptive metadata need itself to be described in and referenced from METS metadata (i.e. the `METS.xml` file) using the element `<dmdSec>` (Figure 12) and as such descriptive metadata are not to be embedded into the METS file directly.

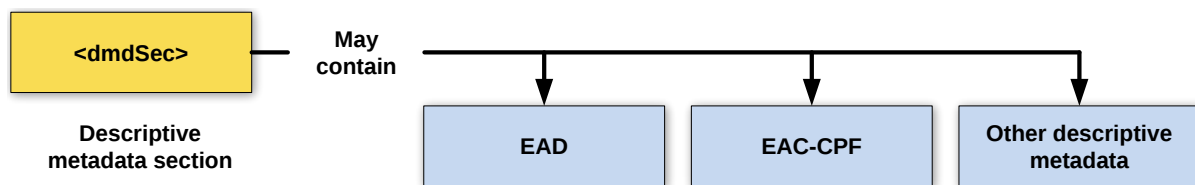


Figure 12: METS descriptive metadata

Following the requirement of explicitly and physically separating descriptive metadata and data we would also like to note, that for interoperability purposes appropriate descriptive metadata elements must explicitly refer to the data content they describe (unless the whole data portion is a single intellectual unit described as a discrete set of descriptive metadata). For example, in the case of EAD elements `<dao>` and `<daogrp>` shall be used to refer to content files from the descriptive metadata. However, regardless of the descriptive metadata standard in question the references from descriptive metadata must always follow the requirement posed in Section 5.1 above (i.e. create references according to the format defined in RFC 3986, or to express references as a relative path to the data files).

Finally we would also note that the recommendation of the CSIP is to always include detailed metadata about intellectual access restrictions and copyright into descriptive metadata (i.e. not into the METS or PREMIS portions of the IP).

Appendices

Appendix A: E-ARK Information Package METS examples

Example 1: Example of a whole METS document describing an information package with no representations

```

<mets:mets OBJID="uuid-4422c185-5407-4918-83b1-7abfa77de182" LABEL="Sample
  CSIP Information Package with no representations" TYPE="Database"
  CONTENTINFORMATIONTYPE="SIARDDK" PROFILE="https://earkcsip.dilcis.eu/
  profile/E-ARK-CSIP.xml" schemaLocation="http://www.loc.gov/METS/ http
  ://www.loc.gov/standards/mets/mets.xsd http://www.w3.org/1999/xlink
  http://www.loc.gov/standards/mets/xlink.xsd https://dilcis.eu/XML/METS/
  CSIPExtensionMETS https://dilcis.eu/XML/METS/CSIPExtensionMETS/
  DILCISExtensionMETS.xsd">
  <mets:metsHdr CREATEDATE="2018-04-24T14:37:49.602+01:00" LASTMODDATE="
    2018-04-24T14:37:49.602+01:00" RECORDSTATUS="NEW" OAISPACKAGETYPE="
    SIP">
    <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
  
```



```

    <mets:name>RODA-in</mets:name>
    <mets:note NOTETYPE="SOFTWARE VERSION">2.1.0-beta.7</mets:note>
  </mets:agent>
</mets:metsHdr>
<mets:dmdSec ID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F" CREATED="
  2018-04-24T14:37:49.609+01:00">
  <mets:mdRef LOCTYPE="URL" MDTYPE="EAD" MDTYPEVERSION="2002" type="
    simple" href="metadata/descriptive/ead2002.xml" SIZE="903" CREATED="
    "2018-04-24T14:37:49.609+01:00" CHECKSUM="
    F24263BF09994749F335E1664DCE0086DB6DCA323FDB6996938BCD28EA9E8153"
    CHECKSUMTYPE="SHA-256">
  </mets:mdRef>
</mets:dmdSec>
<mets:amdSec>
  <mets:digiprovMD ID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F"
    CREATED="2018-04-24T14:37:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" type="simple" href="metadata/preservation/
      premis1.xml" MDTYPE="PREMIS" MDTYPEVERSION="3.0" MIMETYPE="text/
      xml" SIZE="1211" CREATED="2018-04-24T14:37:52.783+01:00" CHECKSUM
      ="8
      aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
      CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
    </mets:mdRef>
  </mets:digiprovMD>
  <mets:digiprovMD ID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3"
    CREATED="2018-04-24T14:47:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" type="simple" href="metadata/preservation/
      premis2.xml" MDTYPE="PREMIS" MDTYPEVERSION="3.0" MIMETYPE="text/
      xml" SIZE="2854" CREATED="2018-04-24T14:37:52.783+01:00" CHECKSUM
      ="
      d1dfa585dcc9d87268069dc58d5e47956434ec3db4087a75a3885d287f15126f"
      CHECKSUMTYPE="SHA-256" LABEL="premis2.xml">
    </mets:mdRef>
  </mets:digiprovMD>
</mets:amdSec>
<mets:fileSec ID="uuid-CA580D47-8C8B-4E91-ABD5-142EBBE15B84">
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H" USE="
    Documentation">
    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A81" MIMETYPE="
      application/vnd.openxmlformats-officedocument.wordprocessingml.
      document" SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
      CHECKSUM="91
      B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"

```

```

        CHECKSUMTYPE="SHA-256">
        <mets:FLocat LOCTYPE="URL" type="simple" href="Documentation/File.
        docx">
        </mets:FLocat>
    </mets:file>
    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A82" MIMETYPE="
    application/vnd.openxmlformats-officedocument.wordprocessingml.
    document" SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
    CHECKSUM="91
    B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
    CHECKSUMTYPE="SHA-256">
        <mets:FLocat LOCTYPE="URL" type="simple" href="Documentation/File2
        .docx">
        </mets:FLocat>
    </mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F" USE="
Schemas">
    <mets:file ID="uuid-A1B7B0DA-E129-48EF-B431-E553F2977FD6" MIMETYPE="
    text/xsd" SIZE="123917" CREATED="2018-04-24T14:37:49.617+01:00"
    CHECKSUM="0
    BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
    CHECKSUMTYPE="SHA-256">
        <mets:FLocat LOCTYPE="URL" type="simple" href="schemas/ead2002.xsd
        ">
        </mets:FLocat>
    </mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86G" USE="
Representations/Submission/Data" CONTENTINFORMATIONTYPE="SIARDDK">
    <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FD" MIMETYPE="
    xml" SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
    CHECKSUM="7176
    A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
    CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-
    F941CBB138B3 uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
        <mets:FLocat LOCTYPE="URL" type="simple" href="representations/
        Submission/Data/SIARD.xml">
        </mets:FLocat>
    </mets:file>
</mets:fileGrp>
</mets:fileSec>
<mets:structMap ID="uuid-1465D250-0A24-4714-9555-5C1211722FB8" TYPE="

```

```

    PHYSICAL" LABEL="CSIP">
  <mets:div ID="uuid-638362BC-65D9-4DA7-9457-5156B3965A18" LABEL="uuid
    -4422c185-5407-4918-83b1-7abfa77de182">
    <mets:div ID="uuid-A4E1C5B6-CD9B-43EF-8F0C-3FD3AB688F81" LABEL="
      Metadata" ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F uuid
      -48C18DD8-2561-4315-AC39-F941CBB138B3" DMDID="uuid-906F4F12-BA52
      -4779-AE2C-178F9206111F">
    </mets:div>
    <mets:div ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86I" LABEL="
      Documentation">
      <mets:fptr FILEID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H">
      </mets:fptr>
    </mets:div>
    <mets:div ID="uuid-26757DC2-4C0F-4431-85B5-5943D1AB5CA3" LABEL="
      Schemas">
      <mets:fptr FILEID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F">
      </mets:fptr>
    </mets:div>
    <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6736" LABEL="
      Representations">
      <mets:fptr FILEID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86G">
      </mets:fptr>
    </mets:div>
  </mets:div>
</mets:structMap>
</mets:mets>

```

Example 2: Example of a whole METS document describing an information package with representations

```

<mets:mets OBJID="uuid-4422c185-5407-4918-83b1-7abfa77de182" LABEL="Sample
  CSIP Information Package with representations" TYPE="Database" PROFILE
  ="https://earkcsip.dilcis.eu/profile/E-ARK-CSIP.xml" schemaLocation="
  http://www.loc.gov/METS/ http://www.loc.gov/standards/mets/mets.xsd
  http://www.w3.org/1999/xlink http://www.loc.gov/standards/mets/xlink.
  xsd https://dilcis.eu/XML/METS/CSIPExtensionMETS https://dilcis.eu/XML/
  METS/CSIPExtensionMETS/DILCISExtensionMETS.xsd">
  <mets:metsHdr CREATEDATE="2018-04-24T14:37:49.602+01:00" LASTMODDATE="
    2018-04-24T14:37:49.602+01:00" RECORDSTATUS="NEW" OAISPACKAGETYPE="
    SIP">
    <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
      <mets:name>RODA-in</mets:name>
      <mets:note NOTETYPE="SOFTWARE VERSION">2.1.0-beta.7</mets:note>
    </mets:agent>
  </mets:metsHdr>
</mets:mets>

```

```
</mets:agent>
</mets:metsHdr>
<mets:dmdSec ID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F" CREATED="
  2018-04-24T14:37:49.609+01:00">
  <mets:mdRef LOCTYPE="URL" MDTYPE="EAD" MDTYPEVERSION="2002" type="
    simple" href="metadata/descriptive/ead2002.xml" SIZE="903" CREATED="
      2018-04-24T14:37:49.609+01:00" CHECKSUM="
        F24263BF09994749F335E1664DCE0086DB6DCA323FDB6996938BCD28EA9E8153"
        CHECKSUMTYPE="SHA-256">
  </mets:mdRef>
</mets:dmdSec>
<mets:amdSec>
  <mets:digiprovMD ID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F"
    CREATED="2018-04-24T14:37:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" type="simple" href="metadata/preservation/
      premis1.xml" MDTYPE="PREMIS" MDTYPEVERSION="3.0" MIMETYPE="text/
        xml" SIZE="1211" CREATED="2018-04-24T14:37:52.783+01:00" CHECKSUM
          ="8
            aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
            CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
    </mets:mdRef>
  </mets:digiprovMD>
  <mets:digiprovMD ID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B3"
    CREATED="2018-04-24T14:47:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" type="simple" href="metadata/preservation/
      premis2.xml" MDTYPE="PREMIS" MDTYPEVERSION="3.0" MIMETYPE="text/
        xml" SIZE="2854" CREATED="2018-04-24T14:37:52.783+01:00" CHECKSUM
          ="
            d1dfa585dcc9d87268069dc58d5e47956434ec3db4087a75a3885d287f15126f"
            CHECKSUMTYPE="SHA-256" LABEL="premis2.xml">
    </mets:mdRef>
  </mets:digiprovMD>
  <mets:digiprovMD ID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943G"
    CREATED="2018-04-24T14:37:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" type="simple" href="metadata/preservation/
      premis3.xml" MDTYPE="PREMIS" MDTYPEVERSION="3.0" MIMETYPE="text/
        xml" SIZE="1211" CREATED="2018-04-24T14:37:52.783+01:00" CHECKSUM
          ="8
            aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
            CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
    </mets:mdRef>
  </mets:digiprovMD>
```

```

<mets:digiprovMD ID="uuid-48C18DD8-2561-4315-AC39-F941CBB138B4"
  CREATED="2018-04-24T14:47:52.783+01:00">
  <mets:mdRef LOCTYPE="URL" type="simple" href="metadata/preservation/
    premis4.xml" MDTYPE="PREMIS" MDTYPEVERSION="3.0" MIMETYPE="text/
    xml" SIZE="2854" CREATED="2018-04-24T14:37:52.783+01:00" CHECKSUM
    ="
    d1dfa585dcc9d87268069dc58d5e47956434ec3db4087a75a3885d287f15126f"
    CHECKSUMTYPE="SHA-256" LABEL="premis2.xml">
  </mets:mdRef>
</mets:digiprovMD>
</mets:amdSec>
<mets:fileSec ID="uuid-CA580D47-8C8B-4E91-ABD5-142EBBE15B84">
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86H" USE="
    Documentation">
    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A81" MIMETYPE="
      application/vnd.openxmlformats-officedocument.wordprocessingml.
      document" SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
      CHECKSUM="91
      B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
      CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="documentation/File.
        docx">
      </mets:FLocat>
    </mets:file>
    <mets:file ID="uuid-0C0049CA-6DE0-4A6D-8699-7975E4046A82" MIMETYPE="
      application/vnd.openxmlformats-officedocument.wordprocessingml.
      document" SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00"
      CHECKSUM="91
      B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557"
      CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" type="simple" href="documentation/File2
        .docx">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F" USE="
    Schemas">
    <mets:file ID="uuid-A1B7B0DA-E129-48EF-B431-E553F2977FD6" MIMETYPE="
      text/xsd" SIZE="123917" CREATED="2018-04-24T14:37:49.617+01:00"
      CHECKSUM="0
      BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
      CHECKSUMTYPE="SHA-256">

```

```

    <mets:FLocat LOCTYPE="URL" type="simple" href="schemas/ead2002.xsd
    ">
  </mets:FLocat>
</mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C277" USE="
  Representations/preingest" CONTENTINFORMATIONTYPE="Access database"
  >
  <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FE" MIMETYPE="
    xml" SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00"
    CHECKSUM="7176
    A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
    CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-
    F941CBB138B3 uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
    <mets:FLocat LOCTYPE="URL" type="simple" href="representations/
      preingest/METS.xml">
  </mets:FLocat>
  </mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C278" USE="
  Representations/submission/data" CONTENTINFORMATIONTYPE="SIARDDK"
  ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F uuid-48C18DD8
  -2561-4315-AC39-F941CBB138B3">
  <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FF" MIMETYPE="
    application/xml" SIZE="1338744" CREATED="2018-04-24T14
    :37:49.617+01:00" CHECKSUM="7176
    A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
    CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-
    F941CBB138B3 uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
    <mets:FLocat LOCTYPE="URL" type="simple" href="representations/
      Submission/METS.xml">
  </mets:FLocat>
  </mets:file>
</mets:fileGrp>
<mets:fileGrp ID="uuid-5811D494-6045-4741-924C-A1CFA340C279" USE="
  Representations/ingest/data" CONTENTINFORMATIONTYPE="SIARD1" ADMID=
  "uuid-9124DA4D-3736-4F69-8355-EB79A22E943G uuid-48C18DD8-2561-4315-
  AC39-F941CBB138B4">
  <mets:file ID="uuid-EE23344D-4F64-40C1-8E18-75839EF661FG" MIMETYPE="
    application/xml" SIZE="1338744" CREATED="2018-04-24T14
    :37:49.617+01:00" CHECKSUM="7176
    A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
    CHECKSUMTYPE="SHA-256" ADMID="uuid-48C18DD8-2561-4315-AC39-

```

```

        F941CBB138B3 uuid-9124DA4D-3736-4F69-8355-EB79A22E943F">
    <mets:FLocat LOCTYPE="URL" type="simple" href="representations/
        ingest/METS.xml">
    </mets:FLocat>
</mets:file>
</mets:fileGrp>
</mets:fileSec>
<mets:structMap ID="uuid-1465D250-0A24-4714-9555-5C1211722FB8" TYPE="
    PHYSICAL" LABEL="CSIP">
    <mets:div ID="uuid-638362BC-65D9-4DA7-9457-5156B3965A18" LABEL="uuid
        -4422c185-5407-4918-83b1-7abfa77de182">
        <mets:div ID="uuid-A4E1C5B6-CD9B-43EF-8F0C-3FD3AB688F81" LABEL="
            Metadata" ADMID="uuid-9124DA4D-3736-4F69-8355-EB79A22E943F uuid
                -48C18DD8-2561-4315-AC39-F941CBB138B3 uuid-9124DA4D-3736-4F69
                    -8355-EB79A22E943G uuid-48C18DD8-2561-4315-AC39-F941CBB138B4"
                DMDID="uuid-906F4F12-BA52-4779-AE2C-178F9206111F">
        </mets:div>
        <mets:div ID="uuid-26757DC2-4C0F-4431-85B5-5943D1AB5CA3" LABEL="
            Schemas" CONTENTIDS="">
            <mets:fptr FILEID="uuid-4ACDC6F3-8A36-4A00-A85F-84A56415E86F">
            </mets:fptr>
        </mets:div>
        <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6737" LABEL="
            representations/preingest">
            <mets:mptr LOCTYPE="URL" type="simple" href="representations/
                preingest/METS.xml" title="uuid-5811D494-6045-4741-924C-
                    A1CFA340C277">
            </mets:mptr>
        </mets:div>
        <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6736" LABEL="
            representations/submission">
            <mets:mptr LOCTYPE="URL" type="simple" href="representations/
                submission/METS.xml" title="uuid-5811D494-6045-4741-924C-
                    A1CFA340C278">
            </mets:mptr>
        </mets:div>
        <mets:div ID="uuid-35CB3341-D731-4AC3-9622-DB8901CD6738" LABEL="
            representations/ingest">
            <mets:mptr LOCTYPE="URL" type="simple" href="representations/
                ingest/METS.xml" title="uuid-5811D494-6045-4741-924C-
                    A1CFA340C279">
            </mets:mptr>
        </mets:div>
    </mets:div>

```

```
</mets:div>
</mets:structMap>
</mets:mets>
```

Appendix B: External Schema and Vocabularies

External Schema

E-ARK CSIP METS Extension

Location: <http://earkcsip.dilcis.eu/schema/DILCISExtensionMETS.xsd> Context: XML-schema for the attributes added by CSIP Note: An extension schema with the added attributes for use in this profile. The schema is identified using the namespace prefix csip.

PREMIS

Location: <http://www.loc.gov/standards/premis/> Context: Used for preservation metadata Note: A rule set for use with this profile is under development.

Controlled Vocabularies

Content information type specification

Maintained By: DILCIS Board Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyContentInformationType.xml> Context: Values for `@csip:CONTENTINFORMATIONTYPE` Description: Lists the names of specific E-ARK content information type specifications supported or maintained in this METS profile.

Content Category

Maintained By: DILCIS Board Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyContentCategory.xml> Context: Values for `mets/@type` Description: Declares the categorical classification of package content.

OAIS Package type

Maintained By: DILCIS Board Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyOAISPackageType.xml> Context: Values for `@csip:OAISPACKAGETYPE` Description: Describes the OAIS type the package belongs to in the OAIS reference model.

Note type

Maintained By: DILCIS Board Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyNoteType.xml>
Context: Values for [@csip:NOTETYPE](#) Description: Provides values for the type of a note for an agent.

Other agent type

Maintained By: DILCIS Board Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyAgentOtherType.xml>
Context: Values for [metsHdr/agent/@OTHERTYPE](#) Description: Describes the other agent types supported by the profile.

Identifier type

Maintained By: Library of Congress Location: <http://id.loc.gov/vocabulary/identifiers.html> Context:
Values for [metsHdr/altRecordID/@TYPE](#) Description: Describes the type of the identifier.

dmdSec status

Maintained By: DILCIS Board Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyStatus.xml>
Context: Values for [dmdSec/@STATUS](#) Description: Describes the status of the descriptive metadata section (dmdSec) which is supported by the profile.

IANA media types

Maintained By: IANAs Location: <https://www.iana.org/assignments/media-types/media-types.xhtml>
Context: Values for [@MIMETYPE](#) Description: Valid values for the mime types of referenced files.

File group names

Maintained By: DILCIS Board Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyFileGrpAndStructMapDivision.xml>
Context: Values for [fileGrp/@USE](#) Description: Describes the uses of the file group `<fileGrp>` that are supported by the profile. Own names should be placed in an own extending vocabulary.

Structural map typing

Maintained By: DILCIS Board Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyStructMapType.xml>

Context: Values for `structMap/@TYPE` Description: Describes the type of the structural map < `structMap`> that is supported by the profile. Own types should be placed in an own extending vocabulary.

Structural map label

Maintained By: DILCIS Board Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyStructMapLabel.xml>

Context: Values for `structMap/@LABEL` Description: Describes the label of the structural map that is supported by the profile. Own labels should be placed in an own extending vocabulary.

Appendix C: A Full List of E-ARK CSIP Requirements

ID	Name & Loc	Description & usage	Card & Level
CSIP1	Package Identifier <code>structMap/@OBJID</code>	The <code>structMap/@OBJID</code> attribute is mandatory, its value is a string identifier for the METS document. For the package METS document, this should be the name/ID of the package, i.e. the name of the package root folder. For a representation level METS document this value records the name/ID of the representation, i.e. the name of the top-level representation folder.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP2	Content Category <code>mets/@TYPE</code>	The <code>mets/@TYPE</code> attribute MUST be used to declare the category of the content held in the package, e.g. book, journal, stereograph, video, etc.. Legal values are defined in a fixed vocabulary. When the content category used falls outside of the defined vocabulary the <code>mets/@TYPE</code> value must be set to “OTHER” and the specific value declared in <code>mets/@csip:OTHERTYPE</code> . The vocabulary will develop under the curation of the DILCIS Board as additional content information type specifications are produced. See also: Content Category	1..1 MUST
CSIP3	Other Content Category <code>mets[@TYPE='OTHER']/@csip:OTHERTYPE</code>	When the <code>mets/@TYPE</code> attribute has the value “OTHER” the <code>mets/@csip:OTHERTYPE</code> attribute MUST be used to declare the content category of the package/representation. See also: Content Category	0..1 SHOULD
CSIP4	Content Information Type Specification <code>mets/@csip:CONTENTINFORMATIONTYPE</code>	Used to declare the Content Information Type Specification used when creating the package. Legal values are defined in a fixed vocabulary. The attribute is mandatory for representation level METS documents. The vocabulary will evolve under the care of the DILCIS Board as additional Content Information Type Specifications are developed. See also: Content information type specification	0..1 SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP5	Other Content Information Type Specification <code>metes[@csip:CONTENTINFORMATI ='OTHER']/@csip : OTHERCONTENTINFO</code>	When the <code>metes/@csip:</code> <code>CONTENTINFORMATIONTYPE</code> has the value “OTHER” the <code>metes/@csip:</code> <code>OTHERCONTENTINFORMATIONTYPE</code> must state the content information type.	0..1 MAY
CSIP6	METS Profile <code>metes/@PROFILE</code>	The URL of the METS profile that the information package conforms with.	1..1 MUST
CSIP117	Package header <code>metes/metesHdr</code>	General element for describing the package.	1..1 MUST
CSIP7	Package creation date <code>metes/metesHdr/ @CREATEDATE</code>	<code>metes/metesHdr/@CREATEDATE</code> records the date the package was created.	1..1 MUST
CSIP8	Package last modification date <code>metes/metesHdr/ @LASTMODDATE</code>	<code>metes/metesHdr/@LASTMODDATE</code> is mandatory when the package has been modified.	0..1 SHOULD
CSIP9	OAIS Package type information <code>metes/ metesHdr/@csip: OAISPACKAGETYPE</code>	<code>metes/metesHdr/@csip:</code> <code>OAISPACKAGETYPE</code> is an additional CSIP attribute that declares the type of the IP. See also: OAIS Package type	1..1 MUST
CSIP10	Agent <code>metes/ metesHdr/agent</code>	A mandatory agent element records the software used to create the package. Other uses of agents may be described in any local implementations that extend the profile.	1..n MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP11	Agent role <code>mets/metsHdr/agent[@ROLE='CREATOR']</code>	The mandatory agent element MUST have a <code>@ROLE</code> attribute with the value “CREATOR”.	1..1 MUST
CSIP12	Agent type <code>mets/metsHdr/agent[@TYPE='OTHER']</code>	The mandatory agent element MUST have a <code>@TYPE</code> attribute with the value “OTHER”.	1..1 MUST
CSIP13	Agent other type <code>mets/metsHdr/agent[@OTHERTYPE='SOFTWARE']</code>	The mandatory agent element MUST have a <code>@OTHERTYPE</code> attribute with the value “SOFTWARE”. See also: Other agent type	1..1 MUST
CSIP14	Agent name <code>mets/metsHdr/agent/name</code>	The mandatory agent’s name element records the name of the software tool used to create the IP.	1..1 MUST
CSIP15	Agent additional information <code>mets/metsHdr/agent/note</code>	The mandatory agent’s note element records the version of the tool used to create the IP.	1..1 MUST
CSIP16	Classification of the agent additional information <code>mets/metsHdr/agent/note[@csip:NOTETYPE='SOFTWARE VERSION']</code>	The mandatory agent element’s note child has a <code>@csip:NOTETYPE</code> attribute with a fixed value of “SOFTWARE VERSION”. See also: Note type	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP17	Descriptive metadata <code> mets/dmdSec</code>	Must be used if descriptive metadata for the package content is available. Each descriptive metadata section (<code><dmdSec></code>) contains a single description and must be repeated for multiple descriptions, when available. It is possible to transfer metadata in a package using just the descriptive metadata section and/or administrative metadata section.	0..n SHOULD
CSIP18	Descriptive metadata identifier <code> mets/dmdSec/@ID</code>	An <code>xml:id</code> identifier for the descriptive metadata section (<code><dmdSec></code>) used for internal package references. It must be unique within the package.	1..1 MUST
CSIP19	Descriptive metadata creation date <code> mets/dmdSec/@CREATED</code>	Creation date of the descriptive metadata in this section.	1..1 MUST
CSIP20	Status of the descriptive metadata <code> mets/dmdSec/@STATUS</code>	Indicates the status of the package using a fixed vocabulary. See also: <code>dmdSec status</code>	0..1 SHOULD
CSIP21	Reference to the document with the descriptive metadata <code> mets/dmdSec/mdRef</code>	Reference to the descriptive metadata file located in the “metadata” section of the IP.	0..1 SHOULD
CSIP22	Type of locator <code> mets/dmdSec/mdRef[@LOCTYPE='URL']</code>	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP23	Type of link <code>mets/dmdSec/ mdRef[@xlink: type='simple']</code>	Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP24	Resource location <code>mets/dmdSec/ mdRef/@xlink: href</code>	The actual location of the resource. This specification recommends recording a URL type filepath in this attribute.	1..1 MUST
CSIP25	Type of metadata <code>mets/dmdSec/ mdRef/@MDTYPE</code>	Specifies the type of metadata in the referenced file. Values are taken from the list provided by the METS.	1..1 MUST
CSIP26	File mime type <code>mets/dmdSec/ mdRef/@MIMETYPE</code>	The IANA mime type of the referenced file. See also: IANA media types	1..1 MUST
CSIP27	File size <code>mets/dmdSec/ mdRef/@SIZE</code>	Size of the referenced file in bytes.	1..1 MUST
CSIP28	File creation date <code>mets/dmdSec/ mdRef/@CREATED</code>	The creation date of the referenced file.	1..1 MUST
CSIP29	File checksum <code>mets/dmdSec/ mdRef/@CHECKSUM</code>	The checksum of the referenced file.	1..1 MUST
CSIP30	File checksum type <code>mets/dmdSec/ mdRef/ @CHECKSUMTYPE</code>	The type of checksum following the value list present in the METS-standard which has been used for calculating the checksum for the referenced file.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP31	Administrative metadata mets/amdSec	If administrative / preservation metadata is available, it must be described using the administrative metadata section (< amdSec >) element. All administrative metadata is present in a single < amdSec > element. It is possible to transfer metadata in a package using just the descriptive metadata section and/or administrative metadata section.	0..1 SHOULD
CSIP32	Digital provenance metadata mets/amdSec/digiprovMD	For recording information about preservation the standard PREMIS is used. It is mandatory to include one < digiprovMD > element for each piece of PREMIS metadata. The use of PREMIS in METS is following the recommendations in 2017 version of PREMIS in METS Guidelines	0..n SHOULD
CSIP33	Digital provenance metadata identifier mets/amdSec/digiprovMD/@ID	An xml:id identifier for the digital provenance metadata section mets/amdSec/digiprovMD used for internal package references. It must be unique within the package.	1..1 MUST
CSIP34	Status of the digital provenance metadata mets/amdSec/digiprovMD/@STATUS	Indicates the status of the package using a fixed vocabulary. See also: dmdSec status	0..1 SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP35	Reference to the document with the digital provenance metdata <code>mets/amdSec/digiprovMD/mdRef</code>	Reference to the digital provenance metadata file stored in the “metadata” section of the IP.	0..1 SHOULD
CSIP36	Type of locator <code>mets/amdSec/digiprovMD/mdRef[@LOCTYPE='URL']</code>	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP37	Type of link <code>mets/amdSec/digiprovMD/mdRef[@xlink:type='simple']</code>	Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP38	Resource location <code>mets/amdSec/digiprovMD/mdRef/@xlink:href</code>	The actual location of the resource. This specification recommends recording a URL type filepath within this attribute.	1..1 MUST
CSIP39	Type of metadata <code>mets/amdSec/digiprovMD/mdRef/@MDTYPE</code>	Specifies the type of metadata in the referenced file. Values are taken from the list provided by the METS.	1..1 MUST
CSIP40	File mime type <code>mets/amdSec/digiprovMD/mdRef/@MIMETYPE</code>	The IANA mime type for the referenced file. See also: IANA media types	1..1 MUST
CSIP41	File size <code>mets/amdSec/digiprovMD/mdRef/@SIZE</code>	Size of the referenced file in bytes.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP42	File creation date <code>mets/amdSec/ digiprovMD/ mdRef/@CREATED</code>	Creation date of the referenced file.	1..1 MUST
CSIP43	File checksum <code>mets/amdSec/ digiprovMD/ mdRef/@CHECKSUM</code>	The checksum of the referenced file.	1..1 MUST
CSIP44	File checksum type <code>mets/amdSec/ digiprovMD/ mdRef/ @CHECKSUMTYPE</code>	The type of checksum following the value list present in the METS-standard which has been used for calculating the checksum for the referenced file.	1..1 MUST
CSIP45	Rights metadata <code>mets/amdSec/ rightsMD</code>	A simple rights statement may be used to describe general permissions for the package. Individual representations should state their specific rights in their representation METS file. Available standards include RightsStatements.org , Europeana rights statements info), METS Rights Schema created and maintained by the METS Board, the rights part of PREMIS as well as own local rights statements in use.	0..n MAY
CSIP46	Rights metadata identifier <code>mets/amdSec/ rightsMD/@ID</code>	An <code>xml:id</code> identifier for the rights metadata section (<code><rightsMD></code>) used for internal package references. It must be unique within the package.	1..1 MUST
CSIP47	Status of the rights metadata <code>mets/ amdSec/rightsMD /@STATUS</code>	Indicates the status of the package using a fixed vocabulary. See also: dmdSec status	0..1 SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP48	Reference to the document with the rights metadata <code>mets/amdSec/</code> <code>rightsMD/mdRef</code>	Reference to the rights metadata file stored in the “metadata” section of the IP.	0..1 SHOULD
CSIP49	Type of locator <code>mets/amdSec/</code> <code>rightsMD/mdRef[</code> <code>@LOCTYPE=URL]</code>	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP50	<code>mets/amdSec/</code> <code>rightsMD/mdRef[</code> <code>@xlink:type='</code> <code>simple']</code>	Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP51	Resource location <code>mets/amdSec/</code> <code>rightsMD/mdRef/</code> <code>@xlink:href</code>	The actual location of the resource. We recommend recording a URL type filepath within this attribute.	1..1 MUST
CSIP52	Type of metadata <code>mets/amdSec/</code> <code>rightsMD/mdRef/</code> <code>@MDTYPE</code>	Specifies the type of metadata in the referenced file. Value is taken from the list provided by the METS.	1..1 MUST
CSIP53	File mime type <code>mets/amdSec/</code> <code>rightsMD/mdRef/</code> <code>@MIMETYPE</code>	The IANA mime type for the referenced file. See also: IANA media types	1..1 MUST
CSIP54	File size <code>mets/</code> <code>amdSec/rightsMD</code> <code>/mdRef/@SIZE</code>	Size of the referenced file in bytes.	1..1 MUST
CSIP55	File creation date <code>mets/amdSec/</code> <code>rightsMD/mdRef/</code> <code>@CREATED</code>	Creation date of the referenced file.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP56	File checksum <code>mets/amdSec/ rightsMD/mdRef/ @CHECKSUM</code>	The checksum of the referenced file.	1..1 MUST
CSIP57	File checksum type <code>mets/amdSec/ rightsMD/mdRef/ @CHECKSUMTYPE</code>	The type of checksum following the value list present in the METS-standard which has been used for calculating the checksum for the referenced file.	1..1 MUST
CSIP58	File section <code>mets/fileSec</code>	The transferred content is placed in the file section in different file group elements, described in other requirements. Only a single file section (<fileSec>) element should be present. It is possible to transfer just descriptive metadata and/or administrative metadata without files placed in this section.	0..1 SHOULD
CSIP59	File section identifier <code>mets/fileSec/@ID</code>	An <code>xml:id</code> identifier for the file section used for internal package references. It must be unique within the package.	1..1 MUST
CSIP60	Documentation file group <code>mets/fileSec/ fileGrp[@USE=' Documentation']</code>	All documentation pertaining to the transferred content is placed in one or more file group elements with <code>mets/fileSec/fileGrp/@USE</code> attribute value “Documentation”. See also: File group names	1..n MUST
CSIP113	Schema file group <code>mets/fileSec/ fileGrp[@USE=' Schemas']</code>	All XML schemas used in the information package should be referenced from one or more file groups with <code>mets/fileSec/fileGrp/@USE</code> attribute value “Schemas”. See also: File group names	1..n MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP114	Representations file group <code> mets/fileSec/ fileGrp[@USE=' Representations ']</code>	A pointer to the METS document describing the representation or pointers to the content being transferred must be present in one or more file groups with <code> mets/fileSec/fileGrp/@USE</code> attribute value “Representations”. See also: File group names	1..n MUST
CSIP61	Reference to administrative metadata <code> mets/fileSec/ fileGrp/@ADMID</code>	If administrative metadata has been provided at file group <code> mets/fileSec/fileGrp</code> level this attribute refers to its administrative metadata section by ID.	0..1 MAY
CSIP62	Content Information Type Specification <code> mets/fileSec/ fileGrp[@LABEL=' Representations ']/@csip: CONTENTINFORMATI</code>	An added attribute which states the name of the content information type specification used to create the package. The vocabulary will evolve under the curation of the DILCIS Board as additional content information type specifications are developed. This attribute is mandatory when the <code> mets/fileSec/fileGrp/@LABEL</code> attribute value is “Representations”. When the “Package type” value is “Mixed” and/or the file group describes a “Representation”, then this element states the content information type specification used for the file group. See also: Content information type specification	0..1 SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP63	Other Content Information Type Specification <code>mets/fileSec/fileGrp[@csip:CONTENTINFORMATIONTYPE= 'OTHER']/@csip:OTHERCONTENTINFORMATIONTYPE</code>	When the <code>mets/fileSec/fileGrp[@csip:CONTENTINFORMATIONTYPE= 'OTHER']</code> attribute has the value “OTHER” the <code>mets/fileSec/fileGrp[@csip:OTHERCONTENTINFORMATIONTYPE</code> must state a value for the Content Information Type Specification used.	0..1 MAY
CSIP64	Description of the use of the file group <code>mets/fileSec/fileGrp/@USE</code>	The value in the <code>mets/fileSec/fileGrp/@USE</code> is the name of the whole folder structure to the data, e.g “Documentation”, “Schemas”, “Representations/preingest” or “Representations/submission/data”.	1..1 MUST
CSIP65	File group identifier <code>mets/fileSec/fileGrp/@ID</code>	An <code>xml:id</code> identifier for the file group used for internal package references. It must be unique within the package.	1..1 MUST
CSIP66	File <code>mets/fileSec/fileGrp/file</code>	The file group (<code><fileGrp></code>) contains the file elements which describe the file objects.	1..n MUST
CSIP67	File identifier <code>mets/fileSec/fileGrp/file/@ID</code>	A unique <code>xml:id</code> identifier for this file across the package.	1..1 MUST
CSIP68	File mimetype <code>mets/fileSec/fileGrp/file/@MIMETYPE</code>	The IANA mime type for the referenced file. See also: IANA media types	1..1 MUST
CSIP69	File size <code>mets/fileSec/fileGrp/file/@SIZE</code>	Size of the referenced file in bytes.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP70	File creation date <code>mets/fileSec/ fileGrp/file/ @CREATED</code>	Creation date of the referenced file.	1..1 MUST
CSIP71	File checksum <code>mets /fileSec/fileGrp/ file/@CHECKSUM</code>	The checksum of the referenced file.	1..1 MUST
CSIP72	File checksum type <code>mets/fileSec/ fileGrp/file/ @CHECKSUMTYPE</code>	The type of checksum following the value list present in the METS-standard which has been used for calculating the checksum for the referenced file.	1..1 MUST
CSIP73	File original identification <code>mets/ fileSec/fileGrp/ file/@OWNERID</code>	If an identifier for the file was supplied by the owner it can be recorded in this attribute.	0..1 MAY
CSIP74	File reference to administrative metadata <code>mets/ fileSec/fileGrp/ file/@ADMID</code>	If administrative metadata has been provided for the file this attribute refers to the file's administrative metadata by ID.	0..1 MAY
CSIP75	File reference to descriptive metadata <code>mets/ fileSec/fileGrp/ file/@DMDID</code>	If descriptive metadata has been provided per file this attribute refers to the file's descriptive metadata by ID.	0..1 MAY
CSIP76	File locator reference <code>mets/ fileSec/fileGrp/ file/FLocat</code>	The location of each external file must be defined by the file location <code><FLocat></code> element using the same rules as for referencing metadata files. All references to files should be made using the XLink href attribute and the file protocol using the relative location of the file.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP77	Type of locator <code>mets/fileSec/ fileGrp/file/ FLocat[@LOCTYPE ='URL']</code>	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP78	Type of link <code>mets/fileSec/ fileGrp/file/ FLocat[@xlink: type='simple']</code>	Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP79	Resource location <code>mets/fileSec/ fileGrp/file/ FLocat/@xlink: href</code>	The actual location of the resource. We recommend recording a URL type filepath within this attribute.	1..1 MUST
CSIP80	Structural description of the package <code>mets/structMap</code>	The structural map <code><structMap></code> element is the only mandatory element in the METS. The <code><structMap></code> in the CSIP describes the highest logical structure of the IP. Each METS file must include ONE structural map <code><structMap></code> element used exactly as described here. Institutions can add their own additional custom structural maps as separate <code><structMap></code> sections.	1..n MUST
CSIP81	Type of structural description <code>mets/structMap[@TYPE ='PHYSICAL']</code>	The <code>mets/structMap/@TYPE</code> attribute must take the value “PHYSICAL” from the vocabulary. See also: Structural map typing	1..1 MUST
CSIP82	Name of the structural description <code>mets/structMap[@LABEL ='CSIP']</code>	The <code>mets/structMap/@LABEL</code> attribute value is set to “CSIP” from the vocabulary. See also: Structural map label	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP83	Structural description identifier <code>mets/structMap[@LABEL='CSIP']/ @ID</code>	An <code>xml:id</code> identifier for the structural description (structMap) used for internal package references. It must be unique within the package.	1..1 MUST
CSIP84	Main structural division <code>mets/structMap[@LABEL='CSIP']/ div</code>	The structural map comprises a single division.	1..1 MUST
CSIP85	Main structural division identifier <code>mets/structMap[@LABEL='CSIP']/ div/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP86	Main structural division label <code>mets/structMap[@LABEL='CSIP']/ div/@LABEL</code>	The package's top-level structural division <code><div></code> element's <code>@LABEL</code> attribute value must be identical to the package identifier, i.e. the same value as the <code>mets/@OBJID</code> attribute.	1..1 MUST
CSIP88	Metadata division <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL='Metadata']</code>	The metadata referenced in the administrative and/or descriptive metadata section is described in the structural map with one sub division. When the transfer consist of only administrative and/or descriptive metadata this is the only sub division that occurs.	1..1 MUST
CSIP89	Metadata division identifier <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL='Metadata']/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP90	Metadata division label <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Metadata']</code>	The metadata division <code><div></code> element's <code>@LABEL</code> attribute value must be "Metadata". See also: File group names	1..1 MUST
CSIP91	Metadata division administrative metadata referencing <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Metadata']/ @ADMID</code>	When there is administrative metadata and the <code>amdSec</code> is present, all administrative metadata MUST be referenced via the administrative sections different identifiers. All of the <code><amdSec></code> identifiers are listed in a single <code>@AMDID</code> using spaces as delimiters.	0..1 SHOULD
CSIP92	Metadata division descriptive metadata referencing <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Metadata']/ @DMDID</code>	When there are descriptive metadata and one or more <code>dmdSec</code> is present, all descriptive metadata MUST be referenced via the descriptive section identifiers. Every <code><dmdSec></code> identifier is listed in a single <code>@DMDID</code> attribute using spaces as delimiters.	0..1 SHOULD
CSIP93	Documentation division <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Documentation']</code>	The documentation referenced in the file section file groups is described in the structural map with one sub division.	0..1 SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP94	Documentation division identifier <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Documentation']/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP95	Documentation division label <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Documentation']</code>	The documentation division <code><div></code> element in the package uses the value “Documentation” from the vocabulary as the value for the <code>@LABEL</code> attribute. See also: File group names	1..1 MUST
CSIP96	Documentation file referencing <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Documentation']/fptr</code>	All file groups containing documentation described in the package are referenced via the relevant file group identifiers. There MUST be one file group reference per <code><fptr></code> element.	0..n MUST
CSIP116	Documentation file group reference pointer <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Documentation']/fptr/@FILEID</code>	A reference, by ID, to the “Documentation” file group. Related to the requirements CSIP60 which describes the “Documentation” file group and CSIP65 which describes the file group identifier.	1..1 MUST
CSIP97	Schema division <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Schemas']</code>	The schemas referenced in the file section file groups are described in the structural map within a single sub-division.	0..1 SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP98	Schema division identifier <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Schemas']/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP99	Schema division label <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Schemas']</code>	The schema division <code><div></code> element's <code>@LABEL</code> attribute has the value “Schemas” from the vocabulary. See also: File group names	1..1 MUST
CSIP100	Schema file reference <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Schemas']/fptr</code>	All file groups containing schemas described in the package are referenced via the relevant file group identifiers. One file group reference per fptr-element	0..n MUST
CSIP118	Schema file group reference <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= 'Schemas']/fptr /@FILEID</code>	The pointer to the identifier for the “Schema” file group. Related to the requirements CSIP113 which describes the “Schema” file group and CSIP65 which describes the file group identifier.	1..1 MUST
CSIP101	Content division <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= ' Representations ']</code>	When no representations are present the content referenced in the file section file group with <code>@USE</code> attribute value “Representations” is described in the structural map as a single sub division.	0..1 SHOULD

ID	Name & Loc	Description & usage	Card & Level
CSIP102	Content division identifier <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= ' Representations ']/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP103	Content division label <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= ' Representations ']</code>	The package's content division <code><div></code> element must have the <code>@LABEL</code> attribute value "Representations", taken from the vocabulary. See also: File group names	1..1 MUST
CSIP104	Content division file references <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= ' Representations ']/fptr</code>	All file groups containing content described in the package are referenced via the relevant file group identifiers. One file group reference per fptr-element	0..n MUST
CSIP119	Content division file group references <code>mets/structMap[@LABEL='CSIP']/ div/div[@LABEL= ' Representations ']/fptr/@FILEID</code>	The pointer to the identifier for the "Representations" file group. Related to the requirements CSIP114 which describes the "Representations" file group and CSIP65 which describes the file group identifier.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP105	Representation division <code>mets/structMap[@LABEL='CSIP']/ div/div</code>	When a package consists of multiple representations, each described by a representation level METS.xml document, there is a discrete representation div element for each representation. Each representation div references the representation level METS.xml document, documenting the structure of the package and its constituent representations.	0..n SHOULD
CSIP106	Representations division identifier <code>mets/structMap[@LABEL='CSIP']/ div/div/@ID</code>	Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP107	Representations division label <code>mets/structMap[@LABEL='CSIP']/ div/div/@LABEL</code>	The package's representation division <code><div></code> element <code>@LABEL</code> attribute value must be the path to the representation level METS document. This requirement gives the same value to be used as the requirement named "File group identifier" (CSIP64) See also: File group names	1..1 MUST
CSIP108	Representations division file referencing <code>mets/structMap[@LABEL='CSIP']/ div/div/mptr/ @xlink:title</code>	The file group containing the files described in the package are referenced via the relevant file group identifier. Related to the requirements CSIP114 which describes the "Representations" file group and CSIP65 which describes the file group identifier.	1..1 MUST

ID	Name & Loc	Description & usage	Card & Level
CSIP109	Representation METS pointer <code>mets/structMap[@LABEL='CSIP']/ div/div/mptr</code>	The division <code><div></code> of the specific representation includes one occurrence of the METS pointer <code><mptr></code> element, pointing to the appropriate representation METS file.	1..1 MUST
CSIP110	Resource location <code>mets/structMap/ div/div/mptr/ @xlink:href</code>	The actual location of the resource. We recommend recording a URL type filepath within this attribute.	1..1 MUST
CSIP111	Type of link <code>mets/ structMap/div/ div/mptr[@xlink :type='simple']</code>	Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP112	Type of locator <code>mets/structMap/ div/div/mptr[@LOCTYPE='URL']</code>	The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
REF_METS_structLink		Section not defined or used in CSIP, additional own uses may occur. Information regarding the structural links is found in the METS Primer	MAY
REF_METS_behaviorSec		Section not defined or used in CSIP, additional own uses may occur. Information regarding the behavior section is found in the METS Primer	MAY

Postface

I Authors

Name	Organisation
Sven Schlarb	Austrian Institute of Technology
Anders Bo Nielsen	Danish National Archives

Name	Organisation
Phillip Mike Tømmerholt	Danish National Archives
Kuldar Aas	Estonian National Archives
Koit Saarevet	Estonian National Archives
Jaime Kaminski	Highbury IVS
Miguel Ferreira	Keep Solutions
Helder Silva	Keep Solutions
Carl Wilson	Open Preservation Foundation
Karin Bredenberg	Swedish National Archives

Contributors to previous version

Name	Organisation
Karin Bredenberg	National Archives of Sweden
Björn Skog	ES Solutions
Anders Bo Nielsen	Danish National Archives
Kathrine Hougaard Edsen Johansen	Danish National Archives
Alex Thirifays	Danish National Archives // Magenta ApS
Sven Schlarb	Austrian Institute of Technology
Andrew Wilson	University of Portsmouth // University of Brighton
Tarvo Kärberg	National Archives of Estonia
Kuldar Aas	National Archives of Estonia
Luis Faria	Keep Solutions
Helder Silva	Keep Solutions
Miguel Ferreira	Keep Solutions
Carl Wilson	Open Preservation Foundation

II Revision History

Revision No.	Date	Authors(s)	Organisation	Description
0.1	17.02.2014	Björn Skog	ESS	First version.
0.2	21.02.2014	Karin Bredenberg	ESS	Updating content.
0.3	24.02.2014	Björn Skog	ESS	Updating content.
0.4	24.10.2014	Tarvo Kärberg	NAE	Updating content.
0.41	05.11.2014	Tarvo Kärberg	NAE	Adding content from Anders Bo Nielsen.
0.42	08.12.2014	Tarvo Kärberg	NAE	Updating content.
0.43	19.12.2014	Tarvo Kärberg	NAE	Updating content.
0.5	26.01.2015	Kathrine Hougaard Edsen	DNA	Updating content.
0.6	11.02.2015	Tarvo Kärberg	NAE	Rearranging content.
0.7	31.05.2015	Kathrine Hougaard Edsen	DNA	Significant changes suggested
0.8	27.07.2015	Tarvo Kärberg	NAE	Merging content
0.9	05.08.2015	Andrew Wilson	UPHEC	Updating content
0.10	07.10.2015	Kuldar Aas	NAE	Major update to include additional details
0.11	30.11.2015	Kuldar Aas	NAE	Intermediate update to include outcomes and decisions from Common Specification meetings
0.12	08.12.2015	Kuldar Aas	NAE	Update on the implementation, include comments from Sven, Jan (AIT) and Andrew (UPHEC)
0.13	05.01.2016	Kuldar Aas, all	NAE, all	Update to include additional comments from E-ARK WPLs and Common Specification group members version sent for external review
0.14	04.07.2016	Kuldar Aas	NAE	Updated structure -> basis for addressing review comments and required updates

Revision No.	Date	Authors(s)	Organisation	Description
0.15	26.08.2016	Kuldar Aas	NAE	Adding available contributions to individual Sections
0.16	05.12.2016	Andrew Wilson, Kuldar Aas	UoB, NAE	Major update. Added section on PREMIS. Revision of tables describing use of METS. General text revisions arising from CS meetings. Updates to requirements.
0.17	16.12.2016	All	All	Final discussions, changes and proofreading before delivering the CS to public comment.
1.0	31.01.2017	Kuldar Aas	NAE	Final small editorial additions
1.1	14.05.2018	Kuldar Aas (editor), DILCIS Board	NAE	Limited proofreading and updates throughout the document. Updates in terminology. Updates in use of METS, ID and referencing section. Improved (more consistent) examples in METS section.
Draft 2.0	28.11.2018	DILCIS Board, E-ARK4ALL	DILCIS Board	Specification updated and released for open review.
2.0	31.05.2019	DILCIS Board, E-ARK4ALL	DILCIS Board	Specification updated after open review.