

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 textual 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.

Preface**I. Aim of the Specification**

This document is one of several related specifications which aim to provide a common set of usage descriptions of international standards for packaging digital information for archiving purposes. These specifications are based on common, international standards for transmitting, describing and preserving digital data. They also utilise the Reference Model for an Open Archival Information System (OAIS), which has Information Packages as its foundation. Familiarity with the core functional entities of OAIS is a prerequisite for understanding the specifications.

The specifications are designed 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. A visualisation of the current specification network can be seen here:

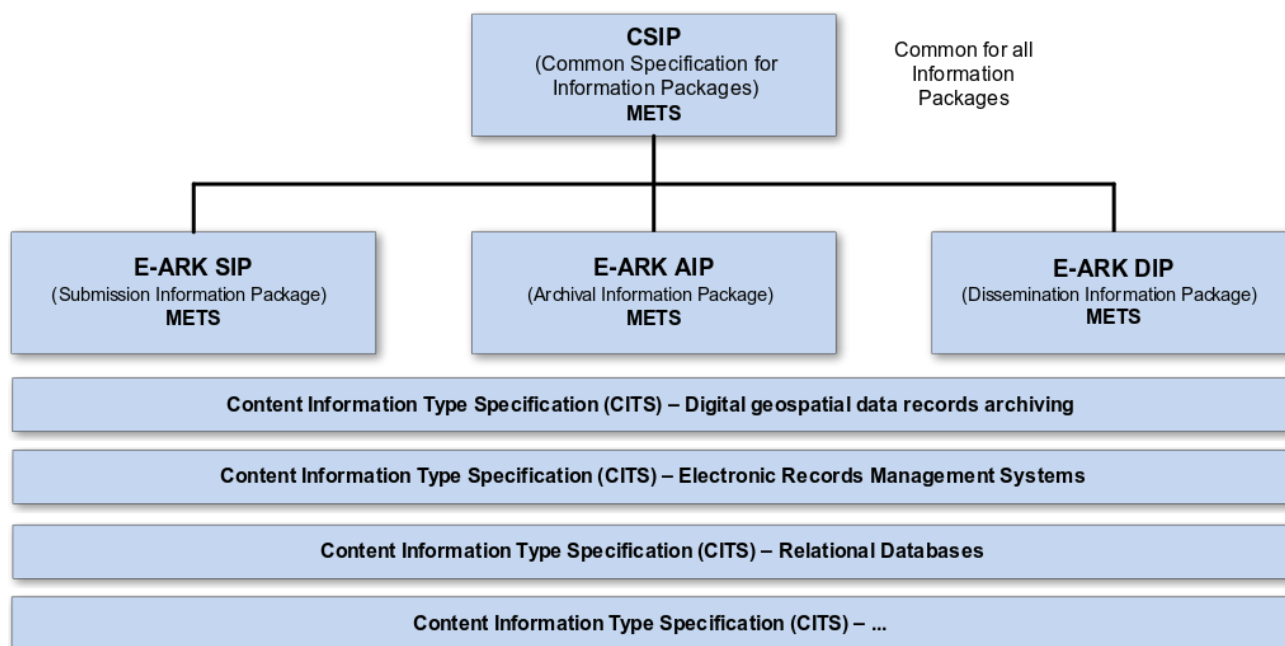


Figure 1: Diagram showing E-ARK specification dependency hierarchy. Note that the image only shows a selection of the published CITS and isn't an exhaustive list.

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). The main purposes of CSIP are to:

- Establish a common understanding of the requirements which need to be met 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, such as document and image collections, databases or geospatial data.
- Enhance interoperability between Producers and Archives.
- Recommend best practices regarding the structure, content and metadata 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 geospatial data.
- Recommend a set of metadata related to the structural and the preservation aspects of the AIP as implemented by the eArchiving 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 aim of a Content Information Type Specification (CITS) is to:

- Define, in technical terms, how data and metadata must be formatted and placed within a CSIP Information Package 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 the DILCIS Board webpage (DILCIS Board, <http://dilcis.eu/>).

II. Organisational Support

This specification is maintained by the Digital Information LifeCycle Interoperability Standards Board (DILCIS Board, <http://dilcis.eu/>). The role of the DILCIS Board is 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 no restriction is placed on the number of participants taking part 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. The DILCIS Board have been responsible for providing the core specifications to 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.

Table of contents

1	Introduction	8
1.1	The Common Specification for Information Packages and OAIS	8
1.2	The Common Specification for Information Packages and Content Information Type Specifications	11
1.3	Common Specification for Information Packages, OAIS Information Packages' specifications and Content Information Type Specifications	12
1.4	Relation to other documents	13
1.4.1	International standards and best-practices	13
1.4.2	Other E-ARK specifications	13
1.5	Structure of the document	14
2	Need for establishing common ground	14
3	Principles for interoperable Information Packages	17
3.1	General principles	18
3.1.1	Principle 1.1	18
3.1.2	Principle 1.2:	18
3.1.3	Principle 1.3	18
3.1.4	Principle 1.4:	19
3.1.5	Principle 1.5:	19
3.1.6	Principle 1.6:	19
3.1.7	Principle 1.7:	19
3.2	Identification of the Information Package	20
3.2.1	Principle 2.1:	20
3.2.2	Principle 2.2:	20
3.2.3	Principle 2.3:	20
3.2.4	Principle 2.4:	20
3.2.5	Principle 2.5:	21
3.3	Structure of the Information Package	21
3.3.1	Principle 3.1:	21
3.3.2	Principle 3.2:	21
3.3.3	Principle 3.3:	22
3.3.4	Principle 3.4:	22
3.3.5	Principle 3.5:	22
3.3.6	Principle 3.6:	23
3.4	Information Package Metadata	24
3.4.1	Principle 4.1:	24
3.4.2	Principle 4.2:	24

3.4.3	Principle 4.3:	24
4	CSIP structure	25
4.1	Folder structure of the CSIP	26
4.2	Implementing the structure	28
5	Use of metadata	29
5.1	General requirements for metadata in a CSIP Information Package	29
5.2	General requirements for the use of metadata	30
5.2.1	The use of identifiers	30
5.2.2	XML Dates and times	31
5.2.3	Referencing between files within a CSIP Information Package	31
5.2.4	Referencing other packages	32
5.2.5	Examples	32
5.3	Use of METS	32
5.3.1	Use of the METS root element (element mets)	33
5.3.2	Use of the METS header (element metsHdr)	36
5.3.3	Use of the METS descriptive metadata section (element dmdSec)	38
5.3.4	Use of the METS administrative metadata section (element amdSec)	40
5.3.5	Use of the METS file section (element fileSec)	45
5.3.6	Use of the METS structural map (element structMap)	52
5.3.7	Deprecated requirements	60
5.4	Use of PREMIS	61
6	Implementation Considerations	61
6.1	Content Information Type Specifications	61
6.1.1	What is a Content Information Type Specification?	61
6.1.2	Maintaining Content Information Type Specifications	62
6.2	Handling large packages	63
6.3	Handling descriptive metadata within the Common Specification	63
7	Appendices	64
7.1	Appendix A: E-ARK Information Package METS examples	64
7.2	Appendix B: External Schema	71
7.2.1	E-ARK CSIP METS Extension	71
7.2.2	PREMIS	71
7.3	Appendix C: External Vocabularies	71
7.3.1	Content information type specification	71
7.3.2	Content Category	71

7.3.3	OAIS Package type	72
7.3.4	Note type	72
7.3.5	Other agent type	72
7.3.6	Identifier type	72
7.3.7	dmdSec status	73
7.3.8	IANA media types	73
7.3.9	File group names	73
7.3.10	Structural map typing	73
7.3.11	Structural map label	74
7.4	Appendix D: A Full List of E-ARK CSIP Requirements	74
7.4.1	Structural Requirements	74
7.4.2	Metadata Requirements	75

1 Introduction

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

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 it to consumers.

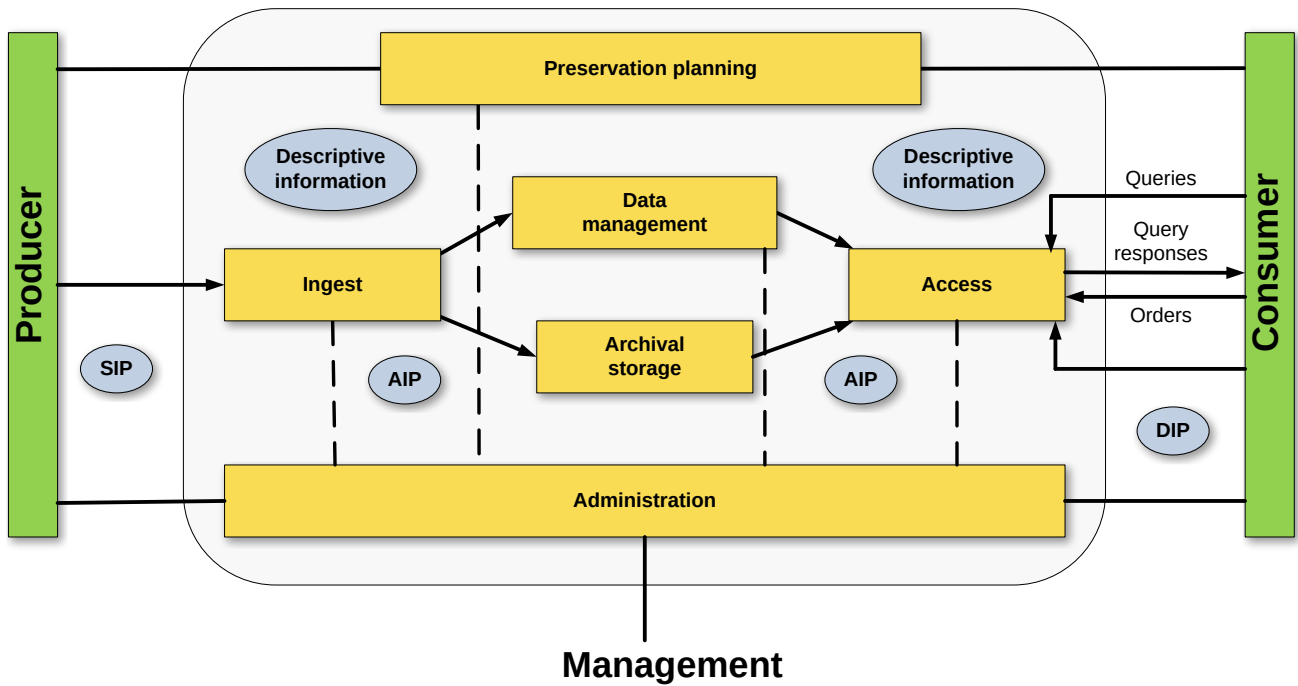


Figure 1: OAIS Functional Entities and Information Packages

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.

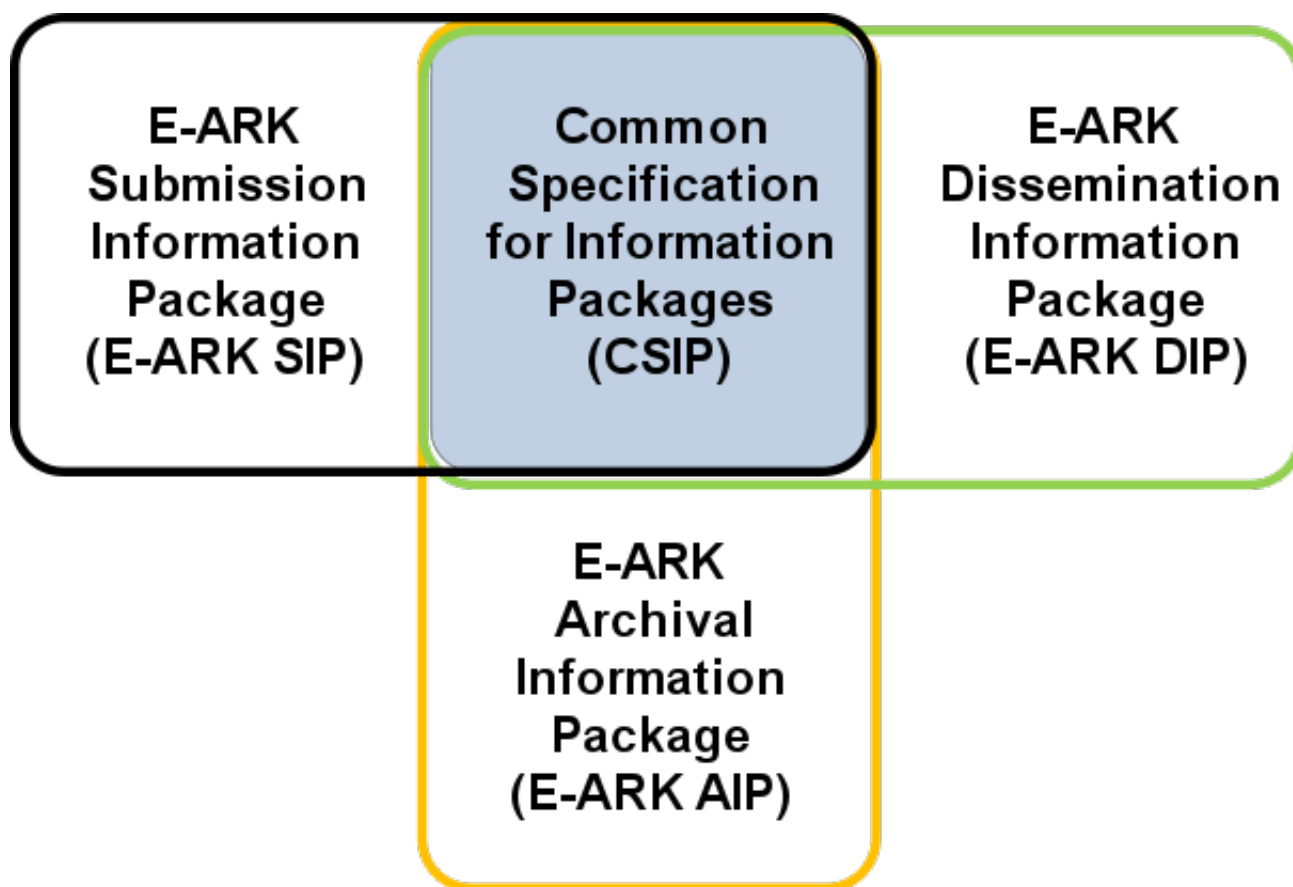


Figure 2: The scope of Common Specification for Information Packages in regard to OAIS Information Packages. In general, the E-ARK SIP and E-ARK DIP specifications reuse and apply fully all the requirements set in this Common Specification. However, they also extend it with aspects relevant only for 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.

It is important to note that the E-ARK AIP format 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 the 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 object’s lifecycle. 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.

E-ARK specifications introduce the concept of Content Information Type Specifications to facilitate such in-depth control over specific content types and formats. 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 May 2019 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 Content Information Type Specification for ERMS: This Content Information Type Specification describes the use of the CSIP for archival records exported from ERMS-type systems (<https://github.com/DILCISBoard/E-ARK-ERMS>);
- E-ARK Content Information Type Specification for Geospatial data: This Content Information Type Specification builds upon the INSPIRE directive and describes the storage of geospatial information within a CSIP compatible Information Package (<https://github.com/DILCISBoard/E-ARK-Geodata>);
- E-ARK Content Information Type Specification for Relational Databases (SIARD1, SIARD2 and SIARDDK): This Content Information Type Specification describes the use of the CSIP for the archiving, preservation and reuse of relational databases using one of the SIARD family (Software Independent Archiving of Relational Databases) formats, (<https://github.com/DILCISBoard/SIARD>). Note, that SIARD1 and SIARDDK specifications are deemed outdated at the time of writing and are only for use when working with existing SIARD1 and SIARDDK packages within CSIP compatible Information Packages. We recommend using the SIARD2 format and the appropriate Content Information Type Specification when creating new archival packages for relational databases.

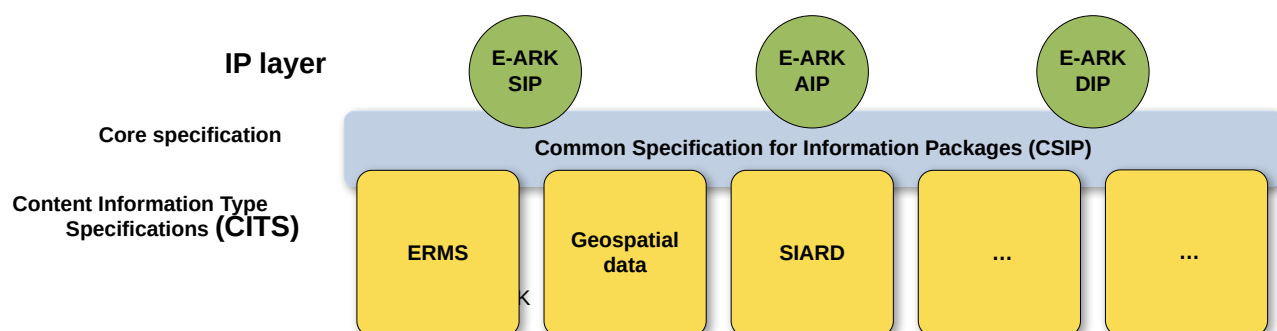


Figure 3: Common Specification for Information Packages and Content Information Type Specifications

The total number of Content Information Type Specifications is 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 see 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

The overall ecosystem of the E-ARK Common Specifications consists of three 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.

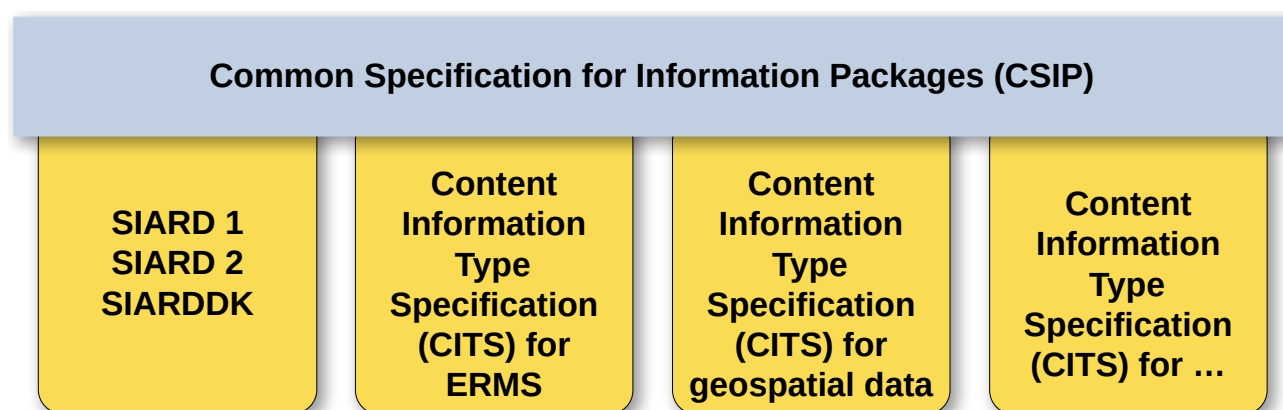


Figure 4: The relationship between the Common Specification for Information Packages; E-ARK SIP, AIP and DIP specifications; and Content Information Type Specifications

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:

1.4.1 International standards and best-practices

- Reference Model for an Open Archival Information System (OAIS), 2012, <https://public.ccsds.org/Pubs/650x0m2.pdf> This specification uses 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) and Dissemination Information Package (DIP).
- Producer-Archive Interface Specification (PAIS) – CCSDS, 2014, <https://public.ccsds.org/Pubs/651x1b1.pdf> The structure of a SIP presented in PAIS has been investigated, but because the implementation of this specification is not very comprehensive yet (only a few prototypes exist), it was decided to rely mainly on the best practices introduced in other reports (see below).

1.4.2 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 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 in the examples directory on this site.

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 reduces 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, it is necessary to consider whether the information needs 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.

Currently, most tools and systems used to create information are not built for coping with long-term requirements for 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).

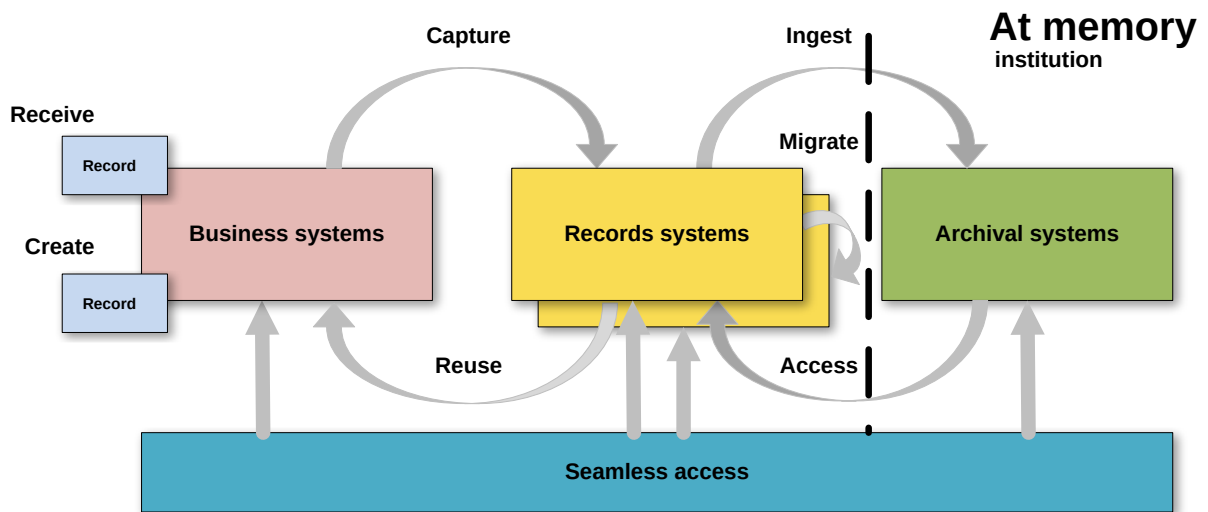


Figure 5: Information flow between live and archival systems

The implication for data owners and system managers is that information which has to be kept for extended time periods need 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 is needed 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, the following interoperability scenarios have been identified (Figure 6):

- Export of data and metadata from source systems and transfer to SIP creation tools (or directly, as a SIP, into preservation systems);
- Transfer of SIPs from SIP creation tools to preservation systems;
- Exchange of preservation systems 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.

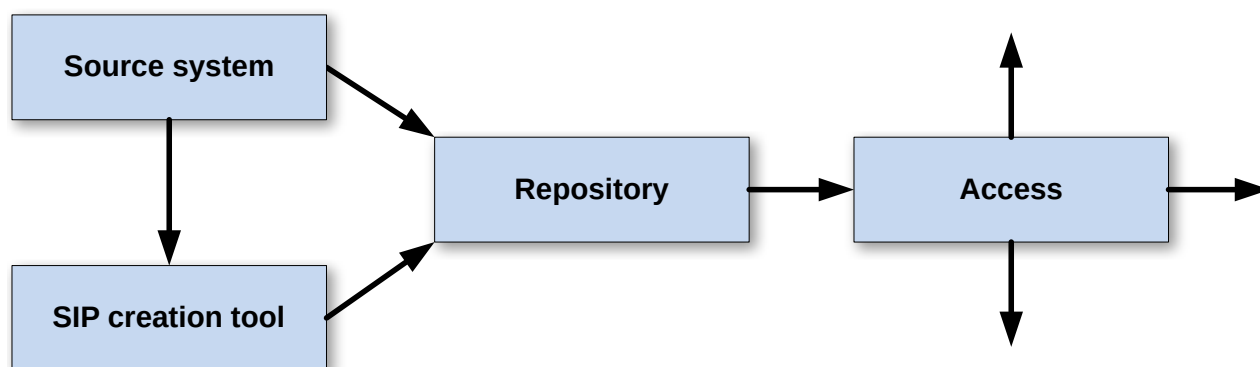


Figure 6: Archival workflow and tool ecosystem

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 generally 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 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 re-developed at each location. Globally, this raises the cost of digital preservation to a level which makes it unaffordable for smaller institutions and, at the same time, does often not allow for developing tools which would be sufficiently mature, user-friendly and not prone to errors. Furthermore, 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, describing 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 leads to a significant decrease in resources needed from any single institution and 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, which is that 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 always includes an RFC2119

keyword (MUST/MUST NOT, SHOULD/SHOULD NOT, MAY) to indicate the requirement level. The short description of each principle is followed by a rationale which describes the reason and background for the principle.

3.1 General principles

3.1.1 Principle 1.1

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

This is one of the most crucial principles of the CSIP. 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.

3.1.2 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. To ensure that a CSIP Information Package is truly interoperable across different platforms, it 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.

3.1.3 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 a 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”.

3.1.4 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 that include 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).

3.1.5 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.

3.1.6 Principle 1.6:

*The Information Package **SHOULD** be human-readable*

In long-term preservation, it is necessary 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.

3.1.7 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

3.2.1 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.

3.2.2 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, it is possible to 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.

3.2.3 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.

3.2.4 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.

3.2.5 Principle 2.5:

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

As stated above, an 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 for achieving an interoperable way of 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 an Information Package are explained in more detail in the following section.

3.3 Structure of the Information Package

3.3.1 Principle 3.1:

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

At the highest level, each Information Package can be logically subdivided 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 the lifetime of a package, this separation also facilitates long-term preservation tasks, such as partial metadata or data updates that do not put the integrity of the package at risk. Regardless of the physical structure of a package, the Information Package **MUST** provide logical separation of data and metadata in the package's manifest.

3.3.2 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.

3.3.3 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 recommended that metadata is divided both logically and physically at least into descriptive and preservation metadata.

3.3.4 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 become obsolete, data and metadata are constantly updated 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.

3.3.5 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 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).

3.3.6 Principle 3.6:

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

Based on principles 3.1 – 3.5, a common structure for any CSIP Information Package is presented (Figure 7).

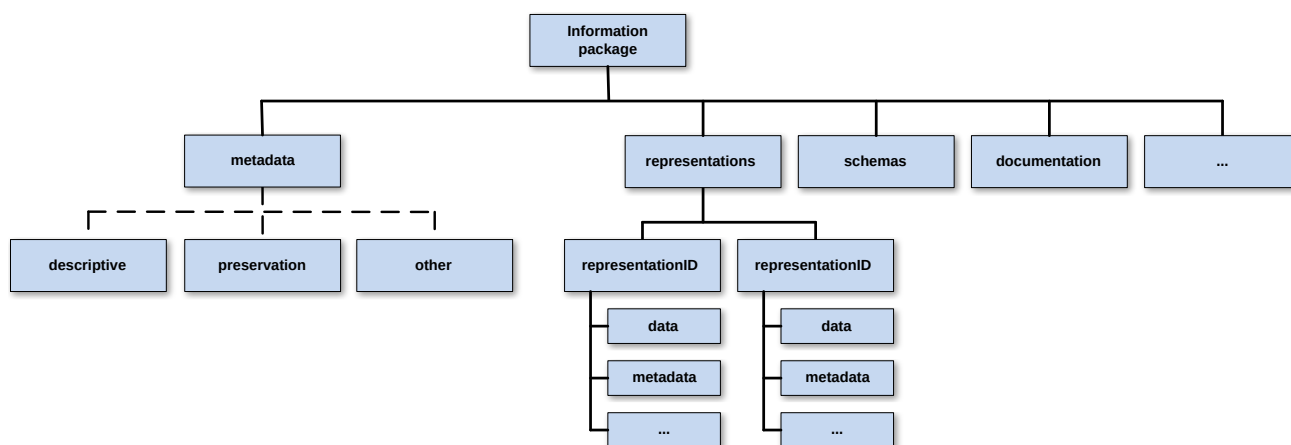


Figure 7: Conceptual structure of the Common Specification

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, it is strongly recommended that this logical structure is manifested as a physical folder structure.

Following Principle 3.3, it is strongly recommended that any package metadata is subdivided into separate metadata sub-types.

Following Principle 3.4, the structure separates the representations of data and metadata explicitly 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

3.4.1 Principle 4.1:

*Metadata in the Information Package **MUST** conform to a standard.*

To exchange, validate, process and reuse Information Packages in an interoperable and automated way, it is necessary 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 of the structure of the package (structural metadata) and the technical details of the data themselves (technical metadata).

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).

3.4.2 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 regarding potential ambiguity. If needed, the selected metadata standard **MUST** be further refined to meet the needs of interoperability and automation.

3.4.3 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. So as not to 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 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 that precisely follows the logical structure. While the specification does not prohibit alternative implementations of the conceptual model, the practice 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. A fixed folder structure, therefore, provides efficiency and scalability.

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 solutions do not support the 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.

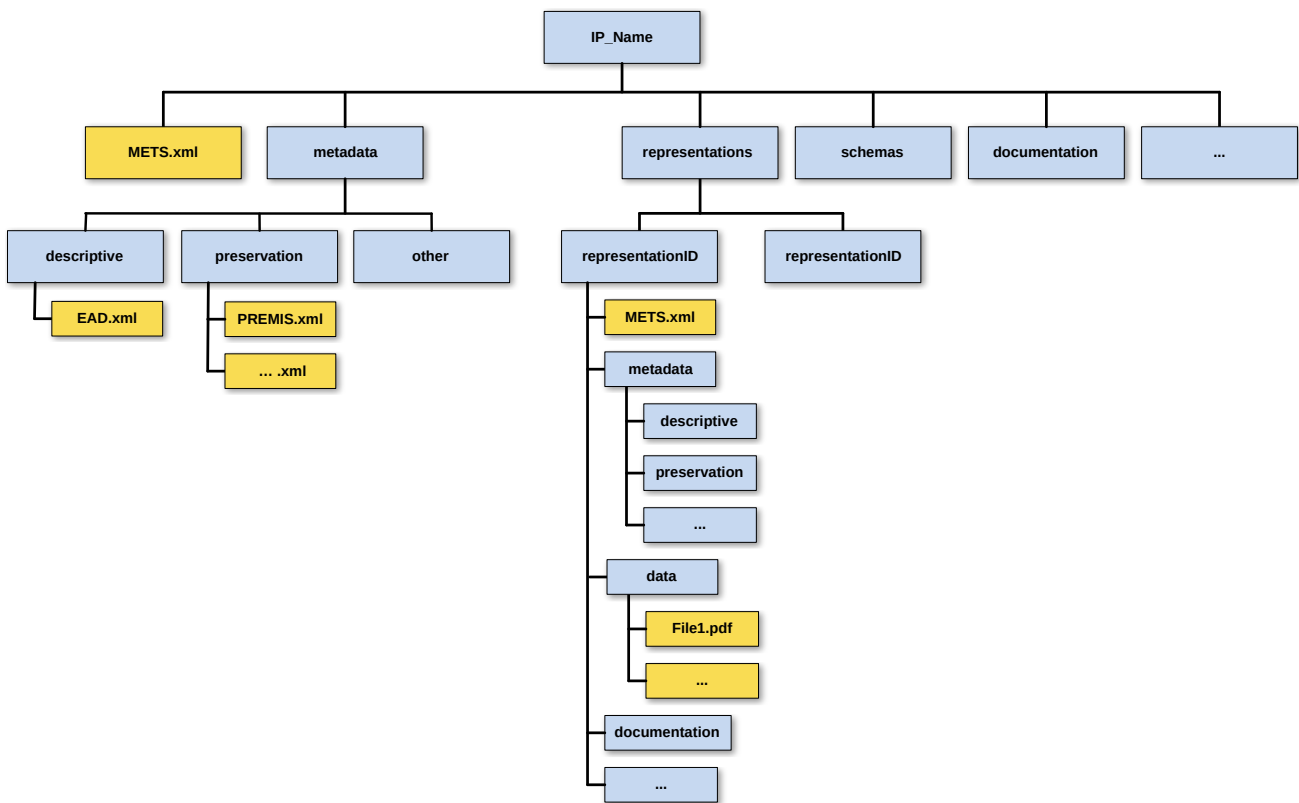


Figure 8: CSIP Information Package folder structure

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 contained 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 **MAY** be contained in an archive/compressed form, e.g. TAR or ZIP, for storage or transfer. The specific format details should be decided by the interested parties and documented, for example in a submission agreement or statement of access terms.

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. Examples of documentation include representation information and manuals for the system the data objects have been exported from.

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.

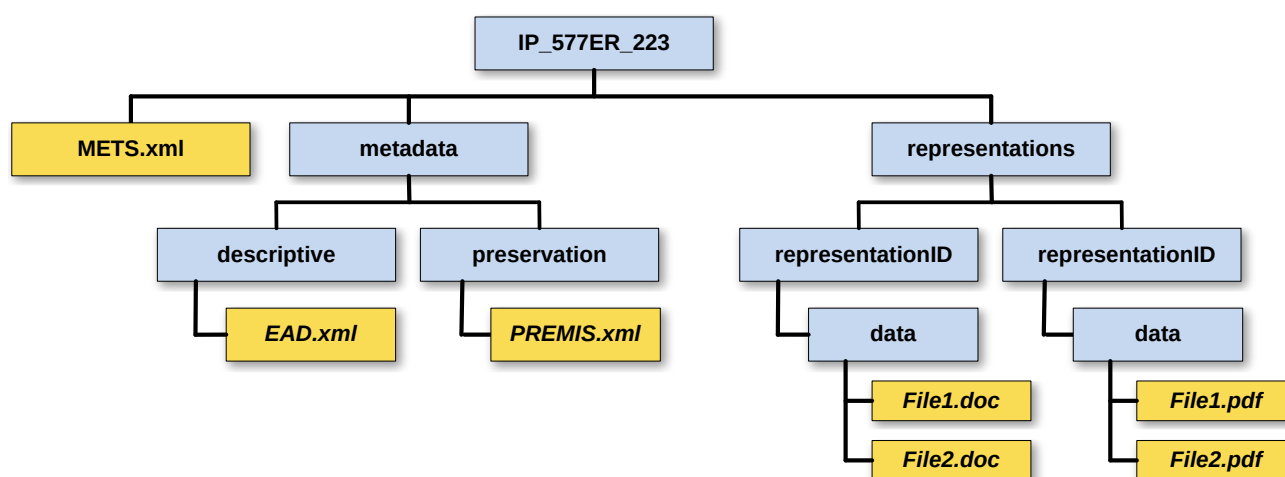


Figure 9: A simple CSIP structure example.

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.

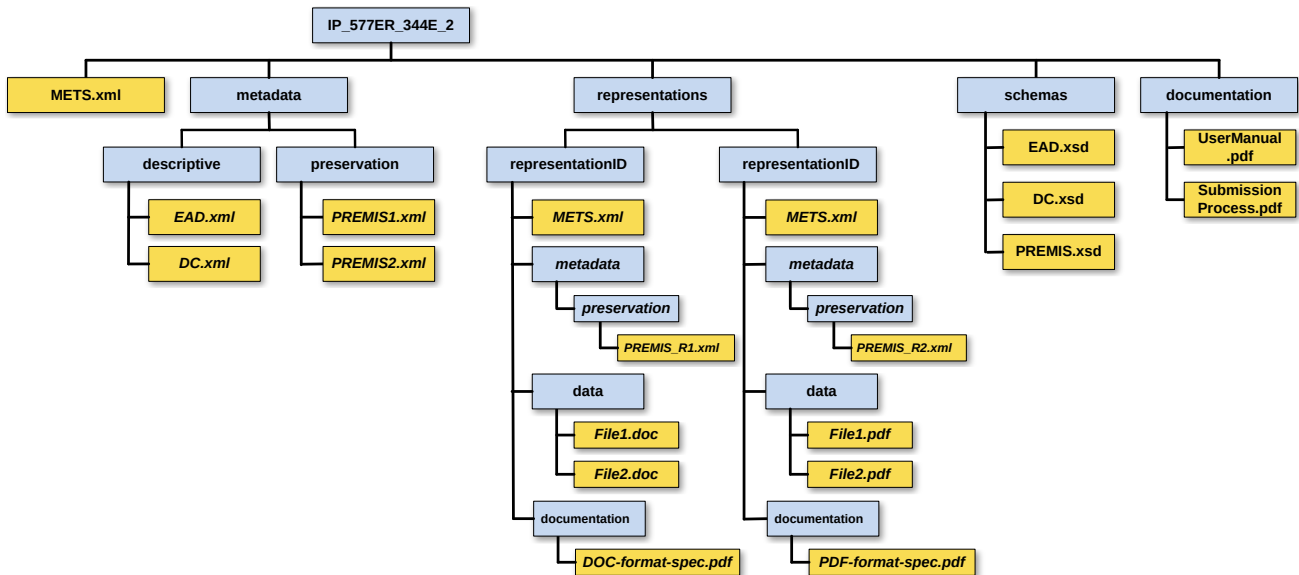


Figure 10: Example of the full use of the CSIP structure

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:

- Uniquely identifying 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);
- validate that no component has been damaged during transfer or preservation (i.e. fixity information);
- understand the processes used when creating and managing the package (i.e. provenance and preservation metadata); and

- understand how the data within the package can be accessed (i.e. representation information).

Descriptive metadata and most detailed technical metadata is outside the scope of the CSIP. As such, the CSIP does not 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.

The core metadata requirements are described 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, it is recognised 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, it is necessary to highlight some general metadata concerns which should be handled consistently.

5.2.1 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 a package makes use of IDs that do not 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 cannot 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">
```

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, it is recommended that a single prefix is chosen and used consistently across all IDs in the package.

Note that while we recommend the use of generated unique IDs in real world information packages we used logical unique names for the examples in this document. This is for readability purposes, particularly for any readers trying to follow reference links.

5.2.2 XML Dates and times

The METS schema prescribes several attributes named `CREATEDATE` OR `LASTMODDATE`, used to record the temporal creation and modification details of package files. While these attributes are simply named `DATE`, they are in fact XML Schema `datetime` and must include a time as well as a date. To use the date and current time of writing as an example, the legal attribute value is `2021-10-12T14:35:10+01:00`, where the `+01:00` denotes the `BST` timezone, one hour in front of Coordinated Universal Time (UTC). In many cases implementers won't have an accurate time, only a date. In these cases simply apply the time of mid-night to the `datetime`, so the previous example expressed as a date would read `2021-10-12T00:00:00+01:00`.

5.2.3 Referencing between files within a CSIP Information Package

It is strongly recommended 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 are not universally supported by all digital preservation tools. To ensure interoperability, all references within a CSIP Information Package must adhere to the requirements stated in this specification.

5.2.4 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.

5.2.5 Examples

This document illustrates the use of the specification with 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, the restrictions imposed by the CSIP on top or those in the official

METS documentation are explained. 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:mets xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:mets="http://www.loc.gov/METS/"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:csip="https://DILCIS.eu/XML/METS/CSIPExtensionMETS"
  xsi: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://
    earkcsip.dilcis.eu/schema/DILCISExtensionMETS.xsd">
```

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, Location & Description	Card & Level
CSIP1	<p>Package Identifier</p> <p><code>mets/@OBJID</code></p> <p>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.</p>	1..1 MUST
CSIP2	<p>Content Category</p> <p><code>mets/@TYPE</code></p> <p>The <code>mets/@TYPE</code> attribute MUST be used to declare the category of the content held in the package, e.g. “Datasets”, “Websites”, “Mixes”, “Other”, 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</p>	1..1 MUST
CSIP3	<p>Other Content Category</p> <p><code>mets[@TYPE='OTHER']/@csip:OTHERTYPE</code></p> <p>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. The value can either be “OTHER” or any other string that are not present in the vocabulary used in the <code>mets/@TYPE</code> attribute.</p>	0..1 SHOULD
CSIP4	<p>Content Information Type Specification</p> <p><code>mets/@csip:CONTENTINFORMATIONTYPE</code></p> <p>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</p>	0..1 SHOULD

ID	Name, Location & Description	Card & Level
CSIP5	<p>Other Content Information Type Specification</p> <p><code>mets[@csip:CONTENTINFORMATIONTYPE='OTHER']/@csip:OTHERCONTENTINFORMATIONTYPE</code></p> <p>When the <code>mets/@csip:CONTENTINFORMATIONTYPE</code> has the value "OTHER" the <code>mets/@csip:OTHERCONTENTINFORMATIONTYPE</code> must state the content information type.</p>	<p>0..1</p> <p>MAY</p>
CSIP6	<p>METS Profile</p> <p><code>mets/@PROFILE</code></p> <p>The URL of the METS profile that the information package conforms with.</p>	<p>1..1</p> <p>MUST</p>

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 xmlns:csip="https://DILCIS.eu/XML/METS/CSIPExtensionMETS" xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink" OBJID="csip-mets-example" LABEL="Sample CSIP Information Package" TYPE="OTHER" csip:OTHERTYPE="Patterns" PROFILE="https://eakcsip.dilcis.eu/profile/E-ARK-CSIP.xml" xsi: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://eakcsip.dilcis.eu/schema/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 xmlns:csip="https://DILCIS.eu/XML/METS/CSIPExtensionMETS" xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink" OBJID="csip-mets-example" LABEL="Sample CSIP Information Package" TYPE="Datasets" csip:CONTENTINFORMATIONTYPE="OTHER" csip:OTHERCONTENTINFORMATIONTYPE="FGS Personal, version 1" PROFILE="https://eakcsip.dilcis.eu/profile/E-ARK-CSIP.xml" xsi:
```

```
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://earkcsip.dilcis.eu/
schema/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, Location & Description	Card & Level
CSIP117	Package header mets/metsHdr General element for describing the package.	1..1 MUST
CSIP7	Package creation datetime mets/metsHdr/@CREATEDATE mets/metsHdr/@CREATEDATE records the date and time the package was created.	1..1 MUST
CSIP8	Package last modification datetime mets/metsHdr/@LASTMODDATE mets/metsHdr/@LASTMODDATE records the data and time the package was modified and is mandatory when the package has been modified.	0..1 SHOULD
CSIP9	OAIS Package type information mets/metsHdr/@csip:OAISPACKAGETYPE mets/metsHdr/@csip:OAISPACKAGETYPE is an additional CSIP attribute that declares the type of the IP. See also: OAIS Package type	1..1 MUST
CSIP10	Agent mets/metsHdr/agent 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, Location & Description	Card & Level
CSIP11	Agent role mets/metsHdr/agent[@ROLE='CREATOR'] The mandatory agent element MUST have a @ROLE attribute with the value "CREATOR".	1..1 MUST
CSIP12	Agent type mets/metsHdr/agent[@TYPE='OTHER'] The mandatory agent element MUST have a @TYPE attribute with the value "OTHER".	1..1 MUST
CSIP13	Agent other type mets/metsHdr/agent[@OTHERTYPE='SOFTWARE'] The mandatory agent element MUST have a @OTHERTYPE attribute with the value "SOFTWARE". See also: Other agent type	1..1 MUST
CSIP14	Agent name mets/metsHdr/agent/name 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 mets/metsHdr/agent/note 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 mets/metsHdr/agent/note[@csip:NOTETYPE='SOFTWARE VERSION'] The mandatory agent element's note child has a @csip:NOTETYPE 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="2018-04-24
T14:37:49.602+01:00" RECORDSTATUS="NEW" csip:OAISPACKAGETYPE="SIP">
  <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
    <mets:name>RODA-in</mets:name>
    <mets:note csip: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 metadata section is to embed or refer to files containing descriptive metadata. CSIP is only using referencing of files containing descriptive metadata.

The CSIP does not prescribe or assume the use of specific descriptive metadata schemas. This means implementers are free to use descriptive metadata standards of their choosing inside a CSIP package following the recommendations in the “E-ARK Content Information Type Specification for Archival Information” (CITS Archival Information) found at <https://citsarchival.dilcis.eu/specification/>.

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

ID	Name, Location & Description	Card & Level
CSIP17	Descriptive metadata mets/dmdSec Must be used if descriptive metadata for the package content is available. Each descriptive metadata section (<dmdSec>) 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 mets/dmdSec/@ID An <code>xml:id</code> identifier for the descriptive metadata section (<dmdSec>) used for internal package references. It must be unique within the package.	1..1 MUST
CSIP19	Descriptive metadata creation datetime mets/dmdSec/@CREATED Creation date and time of the descriptive metadata in this section.	1..1 MUST
CSIP20	Status of the descriptive metadata mets/dmdSec/@STATUS Indicates the status of the package using a fixed vocabulary. See also: dmdSec status	0..1 SHOULD

ID	Name, Location & Description	Card & Level
CSIP21	Reference to the document with the descriptive metadata mets/dmdSec/mdRef Reference to the descriptive metadata file located in the “metadata” section of the IP.	0..1 SHOULD
CSIP22	Type of locator mets/dmdSec/mdRef[@LOCTYPE= 'URL '] The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP23	Type of link mets/dmdSec/mdRef[@xlink:type= 'simple'] Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP24	Resource location mets/dmdSec/mdRef/@xlink:href The actual location of the resource. This specification recommends recording a URL type filepath in this attribute.	1..1 MUST
CSIP25	Type of metadata mets/dmdSec/mdRef/@MDTYPE 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 mets/dmdSec/mdRef/@MIMETYPE The IANA mime type of the referenced file. See also: IANA media types	1..1 MUST
CSIP27	File size mets/dmdSec/mdRef/@SIZE Size of the referenced file in bytes.	1..1 MUST
CSIP28	File creation datetime mets/dmdSec/mdRef/@CREATED The creation date and time of the referenced file.	1..1 MUST
CSIP29	File checksum mets/dmdSec/mdRef/@CHECKSUM The checksum of the referenced file.	1..1 MUST

ID	Name, Location & Description	Card & Level
CSIP30	<p>File checksum type</p> <p>mets/dmdSec/mdRef/@CHECKSUMTYPE</p> <p>A value from the METS-standard which identifies the algorithm used to calculate the checksum for the referenced file.</p>	<p>1..1</p> <p>MUST</p>

Example: METS example of references to descriptive metadata which is described with an EAD document

```
<mets:dmdSec ID="dmd-ead-file" CREATED="2018-04-24T14:37:49.609+01:00">
  <mets:mdRef LOCTYPE="URL" MDTYPE="EAD" xlink:type="simple" xlink: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, Location & Description	Card & Level
CSIP31	Administrative metadata mets/amdSec 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 mets/amdSec/digiprovmD 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 of PREMIS in METS is following the recommendations in the 2017 version of PREMIS in METS Guidelines.	0..n SHOULD
CSIP33	Digital provenance metadata identifier mets/amdSec/digiprovmD/@ID 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 mets/amdSec/digiprovmD/@STATUS Indicates the status of the package using a fixed vocabulary. See also: <code>dmdSec</code> status	0..1 SHOULD
CSIP35	Reference to the document with the digital provenance metadata mets/amdSec/digiprovmD/mdRef Reference to the digital provenance metadata file stored in the “metadata” section of the IP.	0..1 SHOULD

ID	Name, Location & Description	Card & Level
CSIP36	Type of locator mets/amdSec/digiprovmD/mdRef[@LOCTYPE='URL'] The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP37	Type of link mets/amdSec/digiprovmD/mdRef[@xlink:type='simple'] Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP38	Resource location mets/amdSec/digiprovmD/mdRef/@xlink:href The actual location of the resource. This specification recommends recording a URL type filepath within this attribute.	1..1 MUST
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 datetime mets/amdSec/digiprovmD/mdRef/@CREATED Creation date and time 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 A value from the METS-standard which identifies the algorithm used to calculate the checksum for the referenced file.	1..1 MUST

ID	Name, Location & Description	Card & Level
CSIP45	<p>Rights metadata</p> <p>mets/amdSec/rightsMD</p> <p>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.</p>	<p>0..n</p> <p>MAY</p>
CSIP46	<p>Rights metadata identifier</p> <p>mets/amdSec/rightsMD/@ID</p> <p>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.</p>	<p>1..1</p> <p>MUST</p>
CSIP47	<p>Status of the rights metadata</p> <p>mets/amdSec/rightsMD/@STATUS</p> <p>Indicates the status of the package using a fixed vocabulary. See also: dmdSec status</p>	<p>0..1</p> <p>SHOULD</p>
CSIP48	<p>Reference to the document with the rights metadata</p> <p>mets/amdSec/rightsMD/mdRef</p> <p>Reference to the rights metadata file stored in the “metadata” section of the IP.</p>	<p>0..1</p> <p>SHOULD</p>
CSIP49	<p>Type of locator</p> <p>mets/amdSec/rightsMD/mdRef[@LOCTYPE='URL']</p> <p>The locator type is always used with the value “URL” from the vocabulary in the attribute.</p>	<p>1..1</p> <p>MUST</p>
CSIP50	<p>Type of locator</p> <p>mets/amdSec/rightsMD/mdRef[@xlink:type='simple']</p> <p>Attribute used with the value “simple”. Value list is maintained by the xlink standard.</p>	<p>1..1</p> <p>MUST</p>
CSIP51	<p>Resource location</p> <p>mets/amdSec/rightsMD/mdRef/@xlink:href</p> <p>The actual location of the resource. We recommend recording a URL type filepath within this attribute.</p>	<p>1..1</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP52	Type of metadata mets/amdSec/rightsMD/mdRef/@MDTYPE 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 mets/amdSec/rightsMD/mdRef/@MIMETYPE The IANA mime type for the referenced file. See also: IANA media types	1..1 MUST
CSIP54	File size mets/amdSec/rightsMD/mdRef/@SIZE Size of the referenced file in bytes.	1..1 MUST
CSIP55	File creation datetime mets/amdSec/rightsMD/mdRef/@CREATED Creation date and time of the referenced file.	1..1 MUST
CSIP56	File checksum mets/amdSec/rightsMD/mdRef/@CHECKSUM The checksum of the referenced file.	1..1 MUST
CSIP57	File checksum type mets/amdSec/rightsMD/mdRef/@CHECKSUMTYPE A value from the METS-standard which identifies the algorithm used to calculate the checksum for the referenced file.	1..1 MUST

Example: METS example of references to 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="digiprov-premis-file-1" CREATED="2018-04-24T14:37:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" xlink:type="simple" xlink: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="8aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5" CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
  </mets:mdRef>
</mets:digiprovMD>
```

```
< mets: digiprovMD ID="digiprov-premis-file-2" CREATED="2018-04-24T14:47:52.783+01:00">
  < mets: mdRef LOCTYPE="URL" xlink:type="simple" xlink: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, Location & Description	Card & Level
CSIP58	<p>File section mets/fileSec References to all transferred content SHOULD be placed in the file section in the different file group elements, described in other requirements. Only a single file section (<fileSec>) element should be present. In the case that a package only contains metadata updates, i.e. exclusively metadata files, then no file references need to be added to this section.</p>	0..1 SHOULD
CSIP59	<p>File section identifier mets/fileSec/@ID An <code>xml:id</code> identifier for the file section used for internal package references. It must be unique within the package.</p>	1..1 MUST
CSIP60	<p>Documentation file group mets/fileSec/fileGrp[@USE='Documentation'] 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</p>	1..n MUST
CSIP113	<p>Schema file group mets/fileSec/fileGrp[@USE='Schemas'] All XML schemas used in the information package must be referenced from one or more file groups with <code>mets/fileSec/fileGrp/@USE</code> attribute value "Schemas". See also: File group names</p>	1..n MUST
CSIP114	<p>Representations file group mets/fileSec/fileGrp[@USE=[starts-with('Representations')]] 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 starting with "Representations" followed by the path to the folder where the representation level METS document is placed. For example "Representation/submission" and "Representation/ingest". See also: File group names</p>	1..n MUST

ID	Name, Location & Description	Card & Level
CSIP61	<p>Reference to administrative metadata</p> <p><code>mets/fileSec/fileGrp/@ADMID</code></p> <p>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.</p>	<p>0..1</p> <p>MAY</p>
CSIP62	<p>Content Information Type Specification</p> <p><code>mets/@csip:CONTENTINFORMATIONTYPE="MIXED" mets/fileSec/fileGrp[@USE=[starts-with('Representations')]]/@csip:CONTENTINFORMATIONTYPE</code></p> <p>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. When the element “Content Information Type Specification” (CSIP4) has the value “MIXED” or the file group describes a representation, then this element states the content information type specification used for the file group. When the element “Representations file group” (CSIP114), the file group describes a representation with the <code>mets/fileSec/fileGrp/@USE</code> attribute value is starting with “Representations”, then this element must state the content information type specification used for the representation. See also: Content information type specification</p>	<p>0..1</p> <p>SHOULD</p>
CSIP63	<p>Other Content Information Type Specification</p> <p><code>mets/fileSec/fileGrp[@csip:CONTENTINFORMATIONTYPE='OTHER']/@csip:OTHERCONTENTINFORMATIONTYPE</code></p> <p>When the <code>mets/fileSec/fileGrp/@csip:CONTENTINFORMATIONTYPE</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.</p>	<p>0..1</p> <p>MAY</p>
CSIP64	<p>Description of the use of the file group</p> <p><code>mets/fileSec/fileGrp/@USE</code></p> <p>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”.</p>	<p>1..1</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP65	File group identifier mets/fileSec/fileGrp/@ID 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 mets/fileSec/fileGrp/file The file group (<code><fileGrp></code>) contains the file elements which describe the file objects.	1..n MUST
CSIP67	File identifier mets/fileSec/fileGrp/file/@ID A unique <code>xml:id</code> identifier for this file across the package.	1..1 MUST
CSIP68	File mimetype mets/fileSec/fileGrp/file/@MIMETYPE The IANA mime type for the referenced file. See also: IANA media types	1..1 MUST
CSIP69	File size mets/fileSec/fileGrp/file/@SIZE Size of the referenced file in bytes.	1..1 MUST
CSIP70	File creation datetime mets/fileSec/fileGrp/file/@CREATED Creation date and time of the referenced file.	1..1 MUST
CSIP71	File checksum mets/fileSec/fileGrp/file/@CHECKSUM The checksum of the referenced file.	1..1 MUST
CSIP72	File checksum type mets/fileSec/fileGrp/file/@CHECKSUMTYPE A value from the METS-standard which identifies the algorithm used to calculate the checksum for the referenced file.	1..1 MUST
CSIP73	File original identification mets/fileSec/fileGrp/file/@OWNERID If an identifier for the file was supplied by the owner it can be recorded in this attribute.	0..1 MAY

ID	Name, Location & Description	Card & Level
CSIP74	File reference to administrative metadata mets/fileSec/fileGrp/file/@ADMID 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 mets/fileSec/fileGrp/file/@DMDID 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 mets/fileSec/fileGrp/file/FLocat The location of each external file must be defined by the file location <FLocat> element using the same rules as references for 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
CSIP77	Type of locator mets/fileSec/fileGrp/file/FLocat[@LOCTYPE='URL'] The locator type is always used with the value "URL" from the vocabulary in the attribute.	1..1 MUST
CSIP78	Type of link mets/fileSec/fileGrp/file/FLocat[@xlink:type='simple'] Attribute used with the value "simple". Value list is maintained by the xlink standard.	1..1 MUST
CSIP79	Resource location mets/fileSec/fileGrp/file/FLocat/@xlink:href 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="file-sec-example">
  <mets:fileGrp ID="file-grp-doc" USE="Documentation">
    <mets:file ID="file-docx" 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" xlink:type="simple" xlink:href="documentation/
            File.docx">
        </mets:FLocat>
    </mets:file>
    < mets:file ID="file2-docx" 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" xlink:type="simple" xlink:href="documentation/
            File2.docx">
        </mets:FLocat>
    </mets:file>
</mets:fileGrp>
< mets:fileGrp ID="file-grp-schema" USE="Schemas">
    < mets:file ID="file-ead-schema" MIMETYPE="text/xsd" SIZE="123917" CREATED="
        2018-04-24T14:37:49.617+01:00" CHECKSUM="0
        BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
        CHECKSUMTYPE="SHA-256">
        < mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="schemas/ead2002
            .xsd">
        </mets:FLocat>
    </mets:file>
</mets:fileGrp>
< mets:fileGrp ID="file-grp-rep-subdata" USE="Representations/submission/data"
    csip:CONTENTINFORMATIONTYPE="SIARDDK">
    < mets:file ID="file-siard-xml" MIMETYPE="xml" SIZE="1338744" CREATED="
        2018-04-24T14:37:49.617+01:00" CHECKSUM="7176
        A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
        CHECKSUMTYPE="SHA-256" ADMID="digiprovo-premis-file-2 digiprovo-premis-file
        -1">
        < mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink: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="file-sec-example">
  <mets:fileGrp ID="file-grp-doc" USE="Documentation">
    <mets:file ID="file-docx" MIMETYPE="application/vnd.openxmlformats-officedocument.wordprocessingml.document" SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00" CHECKSUM="91B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557" CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="documentation/File.docx">
      </mets:FLocat>
    </mets:file>
    <mets:file ID="file2-docx" MIMETYPE="application/vnd.openxmlformats-officedocument.wordprocessingml.document" SIZE="2554366" CREATED="2012-08-15T12:08:15.432+01:00" CHECKSUM="91B7A2C0A1614AA8F3DAF11DB4A1C981F14BAA25E6A0336F715B7C513E7A1557" CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="documentation/File2.docx">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="file-grp-schema" USE="Schemas">
    <mets:file ID="file-ead-schema" MIMETYPE="text/xsd" SIZE="123917" CREATED="2018-04-24T14:37:49.617+01:00" CHECKSUM="0BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC" CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="schemas/ead2002.xsd">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="file-grp-rep-preing" USE="Representations/preingest" csip:CONTENTINFORMATIONTYPE="OTHER" csip:OTHERCONTENTINFORMATIONTYPE="Access database">
    <mets:file ID="file-preing-mets-xml" MIMETYPE="xml" SIZE="1338744" CREATED="2018-04-24T14:37:49.617+01:00" CHECKSUM="7176A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667" CHECKSUMTYPE="SHA-256" ADMID="digiprov-premis-file-2 digiprov-premis-file-1">
      <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="representations/preingest/METS.xml">

```

```

    </mets:FLocat>
  </mets:file>
</mets:fileGrp>
<mets:fileGrp ID="file-grp-rep-sub" USE="Representations/submission" csip:
  CONTENTINFORMATIONTYPE="SIARDDK" ADMID="digiprov-premis-file-1 digiprov-
  premis-file-2">
  <mets:file ID="file-sub-mets-xml" MIMETYPE="application/xml" SIZE="1338744"
    CREATED="2018-04-24T14:37:49.617+01:00" CHECKSUM="7176
    A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
    CHECKSUMTYPE="SHA-256" ADMID="digiprov-premis-file-2 digiprov-premis-file
    -1">
    <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="representations
      /submission/METS.xml">
    </mets:FLocat>
  </mets:file>
</mets:fileGrp>
<mets:fileGrp ID="file-grp-rep-ing" USE="Representations/ingest" csip:
  CONTENTINFORMATIONTYPE="SIARD1" ADMID="digiprov-premis-file-3 digiprov-
  premis-file-4">
  <mets:file ID="file-ing-mets-xml" MIMETYPE="application/xml" SIZE="1338744"
    CREATED="2018-04-24T14:37:49.617+01:00" CHECKSUM="7176
    A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
    CHECKSUMTYPE="SHA-256" ADMID="digiprov-premis-file-2 digiprov-premis-file
    -1">
    <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="representations
      /ingest/METS.xml">
    </mets:FLocat>
  </mets:file>
</mets:fileGrp>
</mets:fileSec>

```

5.3.6 Use of the METS structural map (element 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.
- 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, Location & Description	Card & Level
CSIP80	<p>Structural description of the package</p> <p><code>mets/structMap</code></p> <p>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. Every CSIP compliant METS file must include ONE structural map <code><structMap></code> element used exactly as described in this section of requirements. Institutions can add their own additional custom structural maps as separate <code><structMap></code> sections following their own requirements.</p>	<p>1..n</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP81	<p>Type of structural description</p> <p>mets/structMap[@TYPE='PHYSICAL']</p> <p>The <code>mets/structMap/@TYPE</code> attribute must take the value “PHYSICAL” from the vocabulary.See also: Structural map typing</p>	<p>1..1</p> <p>MUST</p>
CSIP82	<p>Name of the structural description</p> <p>mets/structMap[@LABEL='CSIP']</p> <p>The <code>mets/structMap/@LABEL</code> attribute value is set to “CSIP” from the vocabulary. This requirement identifies the CSIP compliant structural map <code><structMap></code> element.See also: Structural map label</p>	<p>1..1</p> <p>MUST</p>
CSIP83	<p>Structural description identifier</p> <p>mets/structMap[@LABEL='CSIP']/@ID</p> <p>An <code>xml:id</code> identifier for the structural description (structMap) used for internal package references. It must be unique within the package.</p>	<p>1..1</p> <p>MUST</p>
CSIP84	<p>Main structural division</p> <p>mets/structMap[@LABEL='CSIP']/div</p> <p>The structural map comprises a single division.</p>	<p>1..1</p> <p>MUST</p>
CSIP85	<p>Main structural division identifier</p> <p>mets/structMap[@LABEL='CSIP']/div/@ID</p> <p>Mandatory, <code>xml:id</code> identifier must be unique within the package.</p>	<p>1..1</p> <p>MUST</p>
CSIP88	<p>Metadata division</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Metadata']</p> <p>The metadata referenced in the administrative and/or descriptive metadata section is described in the structural map with one sub division. When the transfer consists of only administrative and/or descriptive metadata this is the only sub division that occurs.</p>	<p>1..1</p> <p>MUST</p>
CSIP89	<p>Metadata division identifier</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Metadata']/@ID</p> <p>Mandatory, <code>xml:id</code> identifier must be unique within the package.</p>	<p>1..1</p> <p>MUST</p>
CSIP90	<p>Metadata division label</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Metadata']</p> <p>The metadata division <code><div></code> element's <code>@LABEL</code> attribute value must be “Metadata”.See also: File group names</p>	<p>1..1</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP91	<p>Metadata division references administrative metadata</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Metadata']/@ADMID</p> <p>The administrative metadata division should reference all current administrative metadata sections. All <amdSec>s with @STATUS='CURRENT' SHOULD be referenced by their identifier, @ID. The current <amdSec> @IDs are recorded in the div[@LABEL='Metadata']/@ADMID attribute in a space delimited list.</p>	<p>0..1</p> <p>SHOULD</p>
CSIP92	<p>Metadata division references descriptive metadata</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Metadata']/@DMDID</p> <p>The descriptive metadata division should reference all current descriptive metadata sections. All <dmdSec>s with @STATUS='CURRENT' SHOULD be referenced by their identifier, @ID. The current <dmdSec> @IDs are recorded in the div[@LABEL='Metadata']/@DMDID attribute in a space delimited list.</p>	<p>0..1</p> <p>SHOULD</p>
CSIP93	<p>Documentation division</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']</p> <p>The documentation referenced in the file section file groups is described in the structural map with one sub division.</p>	<p>0..1</p> <p>SHOULD</p>
CSIP94	<p>Documentation division identifier</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']/@ID</p> <p>Mandatory, xml:id identifier must be unique within the package.</p>	<p>1..1</p> <p>MUST</p>
CSIP95	<p>Documentation division label</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']</p> <p>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</p>	<p>1..1</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP96	Documentation file references mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']/fptr 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 <fptr> element.	0..n MUST
CSIP116	Documentation file group reference pointer mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']/fptr/@FILEID 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 mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas'] 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 mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']/@ID Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP99	Schema division label mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas'] The schema division <div> element’s @LABEL attribute has the value “Schemas” from the vocabulary. See also: File group names	1..1 MUST
CSIP100	Schema file reference mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']/fptr 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

ID	Name, Location & Description	Card & Level
CSIP118	Schema file group reference mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']/fptr/@FILEID 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 mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations'] 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 mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']/@ID Mandatory, <code>xml:id</code> identifier must be unique within the package.	1..1 MUST
CSIP103	Content division label mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations'] The package’s content division <code><div></code> element must have the @LABEL attribute value “Representations”, taken from the vocabulary. See also: File group names	1..1 MUST
CSIP104	Content division file references mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']/fptr 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 mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']/fptr/@FILEID 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, Location & Description	Card & Level
CSIP105	<p>Representation division</p> <p>mets/structMap[@LABEL='CSIP']/div/div</p> <p>When a package consists of multiple representations, each described by a representation level METS.xml document, there should be 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.</p>	<p>0..n</p> <p>SHOULD</p>
CSIP106	<p>Representations division identifier</p> <p>mets/structMap[@LABEL='CSIP']/div/div/@ID</p> <p>Mandatory, <code>xml:id</code> identifier must be unique within the package.</p>	<p>1..1</p> <p>MUST</p>
CSIP107	<p>Representations division label</p> <p>mets/structMap[@LABEL='CSIP']/div/div/@LABEL</p> <p>The package's representation division <code><div></code> element <code>@LABEL</code> attribute value must be the path to the representation level METS document starting with the value "Representations" followed by the main folder name for example "Representations/submission" and "Representations/ingest". This requirement gives the same value to be used as the requirement named "Description of the use of the file group" (CSIP64) See also: File group names</p>	<p>1..1</p> <p>MUST</p>
CSIP108	<p>Representations division file references</p> <p>mets/structMap[@LABEL='CSIP']/div/div/mptr/@xlink:title</p> <p>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.</p>	<p>1..1</p> <p>MUST</p>
CSIP109	<p>Representation METS pointer</p> <p>mets/structMap[@LABEL='CSIP']/div/div/mptr</p> <p>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.</p>	<p>1..1</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP110	Resource location mets/structMap/div/div/mptr/@xlink:href The actual location of the resource. We recommend recording a URL type filepath within this attribute.	1..1 MUST
CSIP111	Type of link mets/structMap/div/div/mptr[@xlink:type='simple'] Attribute used with the value "simple". Value list is maintained by the xlink standard.	1..1 MUST
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="struct-map-example-1" TYPE="PHYSICAL" LABEL="CSIP">
  <mets:div ID="struct-map-example-div" LABEL="csip-mets-example">
    <mets:div ID="struct-map-metadata-div" LABEL="Metadata" ADMID="digipro-
      premis-file-1 digipro-premis-file-2" DMDID="dmd-ead-file">
    </mets:div>
    <mets:div ID="struct-map-doc-div" LABEL="Documentation">
      <mets:fptr FILEID="file-ptr-doc">
      </mets:fptr>
    </mets:div>
    <mets:div ID="struct-map-schema-div" LABEL="Schemas">
      <mets:fptr FILEID="file-grp-schema">
      </mets:fptr>
    </mets:div>
    <mets:div ID="struct-map-reps-sub-div" LABEL="Representations">
      <mets:fptr FILEID="file-grp-rep-subdata">
      </mets:fptr>
    </mets:div>
  </mets:div>
</mets:structMap>
```

Example: METS example of the mandatory structural map including representations

```

<mets:structMap ID="struct-map-example-1" TYPE="PHYSICAL" LABEL="CSIP">
  <mets:div ID="struct-map-example-div" LABEL="csip-mets-example">
    <mets:div ID="struct-map-metadata-div" LABEL="Metadata" ADMID="digiprov-
      premis-file-1 digiprov-premis-file-2 digiprov-premis-file-3 digiprov-
      premis-file-4" DMDID="dmd-ead-file">
    </mets:div>
    <mets:div ID="struct-map-schema-div" LABEL="Schemas">
      <mets:fptr FILEID="file-ptr-schema">
    </mets:fptr>
    </mets:div>
    <mets:div ID="struct-map-reps-preing-div" LABEL="Representations/preingest">
      <mets:mptr LOCTYPE="URL" xlink:type="simple" xlink:href="representations/
        preingest/METS.xml" xlink:title="file-grp-rep-preing">
    </mets:mptr>
    </mets:div>
    <mets:div ID="struct-map-reps-sub-div" LABEL="Representations/submission">
      <mets:mptr LOCTYPE="URL" xlink:type="simple" xlink:href="representations/
        submission/METS.xml" xlink:title="file-grp-rep-sub">
    </mets:mptr>
    </mets:div>
    <mets:div ID="struct-map-reps-ing-div" LABEL="Representations/ingest">
      <mets:mptr LOCTYPE="URL" xlink:type="simple" xlink:href="representations/
        ingest/METS.xml" xlink:title="file-grp-rep-ing">
    </mets:mptr>
    </mets:div>
  </mets:div>
</mets:structMap>

```

5.3.7 Deprecated requirements

The work to improve these specifications is ongoing. On occasion we retire a requirement, these are listed here for information.

ID	Name, Location & Description	Deprecated
CSIP87	Each categorization “Documentation”, “Schemas” as well as each “Representation” within the package must be represented by an occurrence of the division (div) element. Metadata in the administrative and descriptive metadata section has its own division	2019-05-09

ID	Name, Location & Description	Deprecated
CSIP86	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.	2021-10-15

5.4 Use of PREMIS

The CSIP recommends and advocates the use of the PREservation Metadata Implementation Strategies (PREMIS, <https://www.loc.gov/standards/premis/>) metadata standard for recording preservation and technical metadata about digital objects contained within CSIP Information Packages. The use of PREMIS is described in the “E-ARK Content Information Type Specification for Preservation Metadata using PREMIS” (CITS PREMIS) found at <https://citspremis.dilcis.eu/specification/>.

Note that use of PREMIS is not mandatory.

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 the 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 relevant to the specific content type. For example, the ERMS Content Information Type Specification prescribes a method for referencing data (i.e. computer files) from descriptive metadata in ERMS format, ensuring package integrity. 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 Specifications can be domain-specific, and there may 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

It is hoped that many Content Information Type Specifications will be developed with the wider community to create 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 depends on the local implementation. It is also not expected that all tools will support the 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).

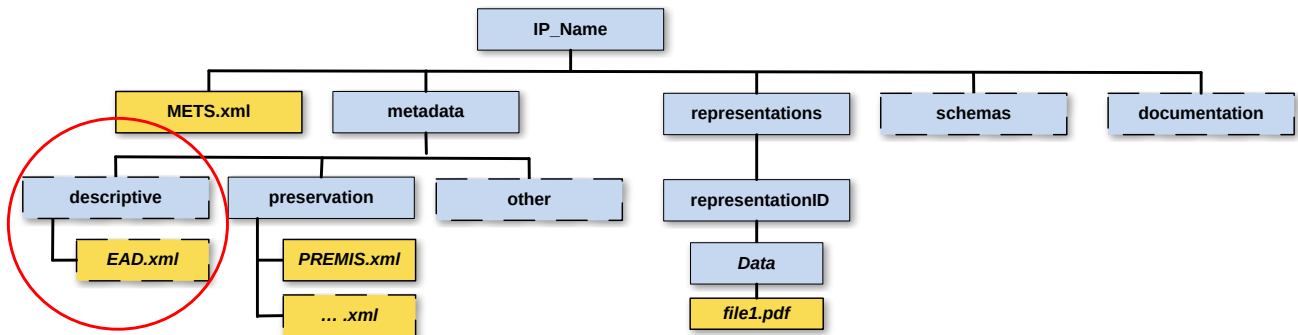


Figure 11: E-ARK IP descriptive metadata

Further, all descriptive metadata need 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 should not be embedded directly in the METS file.

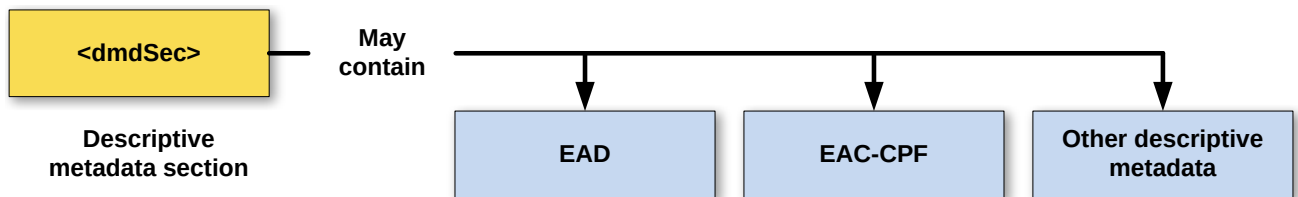


Figure 12: METS descriptive metadata

Following the requirement of explicitly and physically separating descriptive metadata and data it should be noted 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, it is recommended that detailed metadata about intellectual access restrictions and copyright is included in descriptive metadata (i.e. not into the METS or PREMIS portions of the IP).

7 Appendices

7.1 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 xmlns:csip="https://DILCIS.eu/XML/METS/CSIPExtensionMETS" xmlns:mets="
  http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance" xmlns:xlink="http://www.w3.org/1999/xlink" OBJID="csip-mets-example
  " LABEL="Sample CSIP Information Package with no representations" TYPE="
  Database" csip:CONTENTINFORMATIONTYPE="SIARDDK" PROFILE="https://earkcsip.
  dilcis.eu/profile/E-ARK-CSIP.xml" xsi: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://earkcsip.dilcis.eu/schema/DILCISExtensionMETS.xsd">
<mets:metsHdr CREATEDATE="2018-04-24T14:37:49.602+01:00" LASTMODDATE="
  2018-04-24T14:37:49.602+01:00" RECORDSTATUS="NEW" csip:OAISPACKAGETYPE="SIP
  ">
  <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
    <mets:name>RODA-in</mets:name>
    <mets:note csip:NOTETYPE="SOFTWARE VERSION">2.1.0-beta.7</mets:note>
  </mets:agent>
</mets:metsHdr>
<mets:dmdSec ID="appdx1.dmd-ead-file" CREATED="2018-04-24T14:37:49.609+01:00">
  <mets:mdRef LOCTYPE="URL" MDTYPE="EAD" MDTYPEVERSION="2002" xlink:type="
    simple" xlink: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="appdx1.digiprov-premis-file-1" CREATED="2018-04-24T14
:37:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" xlink:type="simple" xlink: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
="8aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
    </mets:mdRef>
  </mets:digiprovMD>
  <mets:digiprovMD ID="appdx1.digiprov-premis-file-2" CREATED="2018-04-24T14
:47:52.783+01:00">
    <mets:mdRef LOCTYPE="URL" xlink:type="simple" xlink: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="appdx1.file-sec-example">
  <mets:fileGrp ID="appdx1.file-grp-doc" USE="Documentation">
    <mets:file ID="appdx1.file-docx" 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" xlink:type="simple" xlink:href="documentation
/File.docx">
      </mets:FLocat>
    </mets:file>
    <mets:file ID="appdx1.file2-docx" 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" xlink:type="simple" xlink:href="documentation
/File2.docx">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
</mets:fileSec>

```

```

    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="appdx1.file-grp-schema" USE="Schemas">
    <mets:file ID="appdx1.file-ead-schema" MIMETYPE="text/xsd" SIZE="123917"
      CREATED="2018-04-24T14:37:49.617+01:00" CHECKSUM="0
      BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
      CHECKSUMTYPE="SHA-256">
      <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="schemas/
        ead2002.xsd">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
  <mets:fileGrp ID="appdx1.file-grp-rep-subdata" USE="Representations/
    Submission/Data" csip:CONTENTINFORMATIONTYPE="SIARDDK">
    <mets:file ID="appdx1.file-siard-xml" MIMETYPE="xml" SIZE="1338744"
      CREATED="2018-04-24T14:37:49.617+01:00" CHECKSUM="7176
      A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
      CHECKSUMTYPE="SHA-256" ADMID="appdx1.digiprov-premis-file-2 appdx1.
      digiprov-premis-file-1">
      <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="
        representations/submission/data/SIARD.xml">
      </mets:FLocat>
    </mets:file>
  </mets:fileGrp>
</mets:fileSec>
<mets:structMap ID="appdx1.struct-map-example-1" TYPE="PHYSICAL" LABEL="CSIP">
  <mets:div ID="appdx1.struct-map-example-div" LABEL="csip-mets-example">
    <mets:div ID="appdx1.struct-map-metadata-div" LABEL="Metadata" ADMID="
      appdx1.digiprov-premis-file-1 appdx1.digiprov-premis-file-2" DMDID="
      appdx1.dmd-ead-file">
    </mets:div>
    <mets:div ID="appdx1.struct-map-doc-div" LABEL="Documentation">
      <mets:fptr FILEID="appdx1.file-grp-doc">
      </mets:fptr>
    </mets:div>
    <mets:div ID="appdx1.struct-map-schema-div" LABEL="Schemas">
      <mets:fptr FILEID="appdx1.file-grp-schema">
      </mets:fptr>
    </mets:div>
    <mets:div ID="appdx1.struct-map-reps-sub-div" LABEL="Representations">
      <mets:fptr FILEID="appdx1.file-grp-rep-subdata">
      </mets:fptr>
    </mets:div>
  </mets:structMap>

```

```

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

```

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

```

<mets:mets xmlns:csip="https://DILCIS.eu/XML/METS/CSIPExtensionMETS" xmlns:mets=
  "http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance" xmlns:xlink="http://www.w3.org/1999/xlink" OBJID="csip-mets-example
  " LABEL="Sample CSIP Information Package with representations" TYPE="Database
  " PROFILE="https://eakcsip.dilcis.eu/profile/E-ARK-CSIP.xml" xsi:
  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://eakcsip.dilcis.eu/
  schema/DILCISExtensionMETS.xsd">
  <mets:metsHdr CREATEDATE="2018-04-24T14:37:49.602+01:00" LASTMODDATE="
    2018-04-24T14:37:49.602+01:00" RECORDSTATUS="NEW" csip:OAISPACKAGETYPE="SIP
    ">
    <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
      <mets:name>RODA-in</mets:name>
      <mets:note csip:NOTETYPE="SOFTWARE VERSION">2.1.0-beta.7</mets:note>
    </mets:agent>
  </mets:metsHdr>
  <mets:dmdSec ID="appdx2.dmd-ead-file" CREATED="2018-04-24T14:37:49.609+01:00">
    <mets:mdRef LOCTYPE="URL" MDTYPE="EAD" MDTYPEVERSION="2002" xlink:type="
      simple" xlink: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="appdx2.digiprov-premis-file-1" CREATED="2018-04-24T14
    :37:52.783+01:00">
      <mets:mdRef LOCTYPE="URL" xlink:type="simple" xlink: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
      ="8aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
      CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
    </mets:mdRef>
  </mets:digiprovMD>

```

```

<mets:digiprovMD ID="appdx2.digiprov-premis-file-2" CREATED="2018-04-24T14
:47:52.783+01:00">
  <mets:mdRef LOCTYPE="URL" xlink:type="simple" xlink: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="appdx2.digiprov-premis-file-3" CREATED="2018-04-24T14
:37:52.783+01:00">
  <mets:mdRef LOCTYPE="URL" xlink:type="simple" xlink: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
  ="8aa278038dbad54bbf142e7d72b493e2598a94946ea1304dc82a79c6b4bac3d5"
  CHECKSUMTYPE="SHA-256" LABEL="premis1.xml">
  </mets:mdRef>
</mets:digiprovMD>
<mets:digiprovMD ID="appdx2.digiprov-premis-file-4" CREATED="2018-04-24T14
:47:52.783+01:00">
  <mets:mdRef LOCTYPE="URL" xlink:type="simple" xlink: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="appdx2.file-sec-example">
  <mets:fileGrp ID="appdx2.file-grp-doc" USE="Documentation">
    <mets:file ID="appdx2.file-docx" 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" xlink:type="simple" xlink:href="documentation
    /File.docx">
    </mets:FLocat>
    </mets:file>
    <mets:file ID="appdx2.file2-docx" 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" xlink:type="simple" xlink:href="documentation
    /File2.docx">
  </mets:FLocat>
</mets:file>
</mets:fileGrp>
<mets:fileGrp ID="appdx2.file-grp-schema" USE="Schemas">
  <mets:file ID="appdx2.file-ead-schema" MIMETYPE="text/xsd" SIZE="123917"
    CREATED="2018-04-24T14:37:49.617+01:00" CHECKSUM="0
    BF9E16ADE296EF277C7B8E5D249D300F1E1EB59F2DCBD89644B676D66F72DCC"
    CHECKSUMTYPE="SHA-256">
    <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="schemas/
      ead2002.xsd">
    </mets:FLocat>
  </mets:file>
</mets:fileGrp>
<mets:fileGrp ID="appdx2.file-grp-rep-preing" USE="Representations/preingest
  " csip:CONTENTINFORMATIONTYPE="OTHER">
  <mets:file ID="appdx2.file-preing-mets-xml" MIMETYPE="xml" SIZE="1338744"
    CREATED="2018-04-24T14:37:49.617+01:00" CHECKSUM="7176
    A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
    CHECKSUMTYPE="SHA-256" ADMID="appdx2.digiprov-premis-file-2 appdx2.
    digiprov-premis-file-1">
    <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="
      representations/preingest/METS.xml">
    </mets:FLocat>
  </mets:file>
</mets:fileGrp>
<mets:fileGrp ID="appdx2.file-grp-rep-sub" USE="Representations/submission"
  csip:CONTENTINFORMATIONTYPE="SIARDDK" ADMID="appdx2.digiprov-premis-file
  -1 appdx2.digiprov-premis-file-2">
  <mets:file ID="appdx2.file-sub-mets-xml" MIMETYPE="application/xml" SIZE="
    1338744" CREATED="2018-04-24T14:37:49.617+01:00" CHECKSUM="7176
    A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
    CHECKSUMTYPE="SHA-256" ADMID="appdx2.digiprov-premis-file-2 appdx2.
    digiprov-premis-file-1">
    <mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="
      representations/submission/METS.xml">
    </mets:FLocat>
  </mets:file>
</mets:fileGrp>
<mets:fileGrp ID="appdx2.file-grp-rep-ing" USE="Representations/ingest" csip
  :CONTENTINFORMATIONTYPE="SIARD1" ADMID="appdx2.digiprov-premis-file-3

```

```

appdx2.digiprov-premis-file-4">
<mets:file ID="appdx2.file-ing-mets-xml" MIMETYPE="application/xml" SIZE="
1338744" CREATED="2018-04-24T14:37:49.617+01:00" CHECKSUM="7176
A627870CFA3854468EC43C5A56F9BD8B30B50A983B8162BF56298A707667"
CHECKSUMTYPE="SHA-256" ADMID="appdx2.digiprov-premis-file-2 appdx2.
digiprov-premis-file-1">
<mets:FLocat LOCTYPE="URL" xlink:type="simple" xlink:href="
representations/ingest/METS.xml">
</mets:FLocat>
</mets:file>
</mets:fileGrp>
</mets:fileSec>
<mets:structMap ID="appdx2.struct-map-example-1" TYPE="PHYSICAL" LABEL="CSIP">
<mets:div ID="appdx2.struct-map-example-div" LABEL="csip-mets-example">
<mets:div ID="appdx2.struct-map-metadata-div" LABEL="Metadata" ADMID="
appdx2.digiprov-premis-file-1 appdx2.digiprov-premis-file-2 appdx2.
digiprov-premis-file-3 appdx2.digiprov-premis-file-4" DMDID="appdx2.dmd
-read-file">
</mets:div>
<mets:div ID="appdx2.struct-map-schema-div" LABEL="Schemas">
<mets:fptr FILEID="appdx2.file-grp-schema">
</mets:fptr>
</mets:div>
<mets:div ID="appdx2.struct-map-reps-preing-div" LABEL="Representations/
preingest">
<mets:mptr LOCTYPE="URL" xlink:type="simple" xlink:href="representations
/preingest/METS.xml" xlink:title="file-grp-rep-preing">
</mets:mptr>
</mets:div>
<mets:div ID="appdx2.struct-map-reps-sub-div" LABEL="Representations/
submission">
<mets:mptr LOCTYPE="URL" xlink:type="simple" xlink:href="representations
/submission/METS.xml" xlink:title="file-grp-rep-sub">
</mets:mptr>
</mets:div>
<mets:div ID="appdx2.struct-map-reps-ing-div" LABEL="Representations/
ingest">
<mets:mptr LOCTYPE="URL" xlink:type="simple" xlink:href="representations
/ingest/METS.xml" xlink:title="file-grp-rep-ing">
</mets:mptr>
</mets:div>
</mets:div>
</mets:structMap>

```

```
</mets:mets>
```

7.2 Appendix B: External Schema

7.2.1 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.

7.2.2 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.

7.3 Appendix C: External Vocabularies

7.3.1 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.

7.3.2 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.

7.3.3 OAIS Package type

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyOAISPackageType.xml>

Context: Values for `@csip:OAISPACKAGE`

Description:

Describes the OAIS type the package belongs to in the OAIS reference model.

7.3.4 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.

7.3.5 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.

7.3.6 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.

7.3.7 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.

7.3.8 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.

7.3.9 File group names

Maintained By: DILCIS Board

Location: <http://earkcsip.dilcis.eu/schema/CSIPVocabularyFileGrpAndStructMapDivisionLabel.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.

7.3.10 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.

7.3.11 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.

7.4 Appendix D: A Full List of E-ARK CSIP Requirements**7.4.1 Structural Requirements**

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 contained 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 **MAY** be contained in an archive/compressed form, e.g. TAR or ZIP, for storage or transfer. The specific format details should be decided by the interested parties and documented, for example in a submission agreement or statement of access terms.

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. Examples of documentation include representation information and manuals for the system the data objects have been exported from.

7.4.2 Metadata Requirements

ID	Name, Location & Description	Card & Level
CSIP1	<p>Package Identifier</p> <p><code>mets/@OBJID</code></p> <p>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.</p>	1..1 MUST
CSIP2	<p>Content Category</p> <p><code>mets/@TYPE</code></p> <p>The <code>mets/@TYPE</code> attribute MUST be used to declare the category of the content held in the package, e.g. “Datasets”, “Websites”, “Mixes”, “Other”, 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</p>	1..1 MUST
CSIP3	<p>Other Content Category</p> <p><code>mets[@TYPE='OTHER']/@csip:OTHERTYPE</code></p> <p>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. The value can either be “OTHER” or any other string that are not present in the vocabulary used in the <code>mets/@TYPE</code> attribute.</p>	0..1 SHOULD
CSIP4	<p>Content Information Type Specification</p> <p><code>mets/@csip:CONTENTINFORMATIONTYPE</code></p> <p>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</p>	0..1 SHOULD

ID	Name, Location & Description	Card & Level
CSIP5	<p>Other Content Information Type Specification</p> <p><code>mets[@csip:CONTENTINFORMATIONTYPE='OTHER']/@csip:OTHERCONTENTINFORMATIONTYPE</code></p> <p>When the <code>mets/@csip:CONTENTINFORMATIONTYPE</code> has the value "OTHER" the <code>mets/@csip:OTHERCONTENTINFORMATIONTYPE</code> must state the content information type.</p>	0..1 MAY
CSIP6	<p>METS Profile</p> <p><code>mets/@PROFILE</code></p> <p>The URL of the METS profile that the information package conforms with.</p>	1..1 MUST
CSIP117	<p>Package header</p> <p><code>mets/metsHdr</code></p> <p>General element for describing the package.</p>	1..1 MUST
CSIP7	<p>Package creation datetime</p> <p><code>mets/metsHdr/@CREATEDATE</code></p> <p><code>mets/metsHdr/@CREATEDATE</code> records the date and time the package was created.</p>	1..1 MUST
CSIP8	<p>Package last modification datetime</p> <p><code>mets/metsHdr/@LASTMODDATE</code></p> <p><code>mets/metsHdr/@LASTMODDATE</code> records the data and time the package was modified and is mandatory when the package has been modified.</p>	0..1 SHOULD
CSIP9	<p>OAIS Package type information</p> <p><code>mets/metsHdr/@csip:OAISPACKAGETYPE</code></p> <p><code>mets/metsHdr/@csip:OAISPACKAGETYPE</code> is an additional CSIP attribute that declares the type of the IP. See also: OAIS Package type</p>	1..1 MUST
CSIP10	<p>Agent</p> <p><code>mets/metsHdr/agent</code></p> <p>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.</p>	1..n MUST
CSIP11	<p>Agent role</p> <p><code>mets/metsHdr/agent[@ROLE='CREATOR']</code></p> <p>The mandatory agent element MUST have a <code>@ROLE</code> attribute with the value "CREATOR".</p>	1..1 MUST

ID	Name, Location & Description	Card & Level
CSIP12	Agent type mets/metsHdr/agent[@TYPE='OTHER'] The mandatory agent element MUST have a @TYPE attribute with the value "OTHER".	1..1 MUST
CSIP13	Agent other type mets/metsHdr/agent[@OTHERTYPE='SOFTWARE'] The mandatory agent element MUST have a @OTHERTYPE attribute with the value "SOFTWARE". See also: Other agent type	1..1 MUST
CSIP14	Agent name mets/metsHdr/agent/name 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 mets/metsHdr/agent/note 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 mets/metsHdr/agent/note[@csip:NOTETYPE='SOFTWARE VERSION'] The mandatory agent element's note child has a @csip:NOTETYPE attribute with a fixed value of "SOFTWARE VERSION". See also: Note type	1..1 MUST
CSIP17	Descriptive metadata mets/dmdSec Must be used if descriptive metadata for the package content is available. Each descriptive metadata section (<dmdSec>) 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 mets/dmdSec/@ID An xml:id identifier for the descriptive metadata section (<dmdSec>) used for internal package references. It must be unique within the package.	1..1 MUST

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

ID	Name, Location & Description	Card & Level
CSIP28	File creation datetime mets/dmdSec/mdRef/@CREATED The creation date and time of the referenced file.	1..1 MUST
CSIP29	File checksum mets/dmdSec/mdRef/@CHECKSUM The checksum of the referenced file.	1..1 MUST
CSIP30	File checksum type mets/dmdSec/mdRef/@CHECKSUMTYPE A value from the METS-standard which identifies the algorithm used to calculate the checksum for the referenced file.	1..1 MUST
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 the 2017 version of PREMIS in METS Guidelines.	0..n SHOULD
CSIP33	Digital provenance metadata identifier mets/amdSec/digiprovMD/@ID 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 mets/amdSec/digiprovMD/@STATUS Indicates the status of the package using a fixed vocabulary. See also: dmdSec status	0..1 SHOULD

ID	Name, Location & Description	Card & Level
CSIP35	Reference to the document with the digital provenance metadata mets/amdSec/digiprovmD/mdRef Reference to the digital provenance metadata file stored in the “metadata” section of the IP.	0..1 SHOULD
CSIP36	Type of locator mets/amdSec/digiprovmD/mdRef[@LOCTYPE='URL'] The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP37	Type of link mets/amdSec/digiprovmD/mdRef[@xlink:type='simple'] Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
CSIP38	Resource location mets/amdSec/digiprovmD/mdRef/@xlink:href The actual location of the resource. This specification recommends recording a URL type filepath within this attribute.	1..1 MUST
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 datetime mets/amdSec/digiprovmD/mdRef/@CREATED Creation date and time of the referenced file.	1..1 MUST
CSIP43	File checksum mets/amdSec/digiprovmD/mdRef/@CHECKSUM The checksum of the referenced file.	1..1 MUST

ID	Name, Location & Description	Card & Level
CSIP44	File checksum type mets/amdSec/digiprovd/mdRef/@CHECKSUMTYPE A value from the METS-standard which identifies the algorithm used to calculate the checksum for the referenced file.	1..1 MUST
CSIP45	Rights metadata mets/amdSec/rightsMD 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 mets/amdSec/rightsMD/@ID 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 mets/amdSec/rightsMD/@STATUS 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 mets/amdSec/rightsMD/mdRef Reference to the rights metadata file stored in the “metadata” section of the IP.	0..1 SHOULD
CSIP49	Type of locator mets/amdSec/rightsMD/mdRef[@LOCTYPE='URL'] The locator type is always used with the value “URL” from the vocabulary in the attribute.	1..1 MUST
CSIP50	Type of locator mets/amdSec/rightsMD/mdRef[@xlink:type='simple'] Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST

ID	Name, Location & Description	Card & Level
CSIP51	Resource location mets/amdSec/rightsMD/mdRef/@xlink:href The actual location of the resource. We recommend recording a URL type filepath within this attribute.	1..1 MUST
CSIP52	Type of metadata mets/amdSec/rightsMD/mdRef/@MDTYPE 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 mets/amdSec/rightsMD/mdRef/@MIMETYPE The IANA mime type for the referenced file. See also: IANA media types	1..1 MUST
CSIP54	File size mets/amdSec/rightsMD/mdRef/@SIZE Size of the referenced file in bytes.	1..1 MUST
CSIP55	File creation datetime mets/amdSec/rightsMD/mdRef/@CREATED Creation date and time of the referenced file.	1..1 MUST
CSIP56	File checksum mets/amdSec/rightsMD/mdRef/@CHECKSUM The checksum of the referenced file.	1..1 MUST
CSIP57	File checksum type mets/amdSec/rightsMD/mdRef/@CHECKSUMTYPE A value from the METS-standard which identifies the algorithm used to calculate the checksum for the referenced file.	1..1 MUST
CSIP58	File section mets/fileSec References to all transferred content SHOULD be placed in the file section in the different file group elements, described in other requirements. Only a single file section (<fileSec>) element should be present. In the case that a package only contains metadata updates, i.e. exclusively metadata files, then no file references need to be added to this section.	0..1 SHOULD

ID	Name, Location & Description	Card & Level
CSIP59	<p>File section identifier</p> <p><code>mets/fileSec/@ID</code></p> <p>An <code>xml:id</code> identifier for the file section used for internal package references. It must be unique within the package.</p>	1..1 MUST
CSIP60	<p>Documentation file group</p> <p><code>mets/fileSec/fileGrp[@USE='Documentation']</code></p> <p>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</p>	1..n MUST
CSIP113	<p>Schema file group</p> <p><code>mets/fileSec/fileGrp[@USE='Schemas']</code></p> <p>All XML schemas used in the information package must be referenced from one or more file groups with <code>mets/fileSec/fileGrp/@USE</code> attribute value "Schemas". See also: File group names</p>	1..n MUST
CSIP114	<p>Representations file group</p> <p><code>mets/fileSec/fileGrp[@USE=[starts-with('Representations')]]</code></p> <p>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 starting with "Representations" followed by the path to the folder where the representation level METS document is placed. For example "Representation/submission" and "Representation/ingest". See also: File group names</p>	1..n MUST
CSIP61	<p>Reference to administrative metadata</p> <p><code>mets/fileSec/fileGrp/@ADMID</code></p> <p>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.</p>	0..1 MAY

ID	Name, Location & Description	Card & Level
CSIP62	<p>Content Information Type Specification</p> <p><code>mets/@csip:CONTENTINFORMATIONTYPE="MIXED" mets/fileSec/fileGrp[@USE=[starts-with('Representations')]]/@csip:CONTENTINFORMATIONTYPE</code></p> <p>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. When the element “Content Information Type Specification” (CSIP4) has the value “MIXED” or the file group describes a representation, then this element states the content information type specification used for the file group. When the element “Representations file group” (CSIP114), the file group describes a representation with the <code>mets/fileSec/fileGrp/@USE</code> attribute value is starting with “Representations”, then this element must state the content information type specification used for the representation. See also: Content information type specification</p>	<p>0..1</p> <p>SHOULD</p>
CSIP63	<p>Other Content Information Type Specification</p> <p><code>mets/fileSec/fileGrp[@csip:CONTENTINFORMATIONTYPE='OTHER']/@csip:OTHERCONTENTINFORMATIONTYPE</code></p> <p>When the <code>mets/fileSec/fileGrp/@csip:CONTENTINFORMATIONTYPE</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.</p>	<p>0..1</p> <p>MAY</p>
CSIP64	<p>Description of the use of the file group</p> <p><code>mets/fileSec/fileGrp/@USE</code></p> <p>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”.</p>	<p>1..1</p> <p>MUST</p>
CSIP65	<p>File group identifier</p> <p><code>mets/fileSec/fileGrp/@ID</code></p> <p>An <code>xml:id</code> identifier for the file group used for internal package references. It must be unique within the package.</p>	<p>1..1</p> <p>MUST</p>
CSIP66	<p>File</p> <p><code>mets/fileSec/fileGrp/file</code></p> <p>The file group (<code><fileGrp></code>) contains the file elements which describe the file objects.</p>	<p>1..n</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP67	File identifier mets/fileSec/fileGrp/file/@ID A unique <code>xml:id</code> identifier for this file across the package.	1..1 MUST
CSIP68	File mimetype mets/fileSec/fileGrp/file/@MIMETYPE The IANA mime type for the referenced file. See also: IANA media types	1..1 MUST
CSIP69	File size mets/fileSec/fileGrp/file/@SIZE Size of the referenced file in bytes.	1..1 MUST
CSIP70	File creation datetime mets/fileSec/fileGrp/file/@CREATED Creation date and time of the referenced file.	1..1 MUST
CSIP71	File checksum mets/fileSec/fileGrp/file/@CHECKSUM The checksum of the referenced file.	1..1 MUST
CSIP72	File checksum type mets/fileSec/fileGrp/file/@CHECKSUMTYPE A value from the METS-standard which identifies the algorithm used to calculate the checksum for the referenced file.	1..1 MUST
CSIP73	File original identification mets/fileSec/fileGrp/file/@OWNERID 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 mets/fileSec/fileGrp/file/@ADMID 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 mets/fileSec/fileGrp/file/@DMDID If descriptive metadata has been provided per file this attribute refers to the file's descriptive metadata by ID.	0..1 MAY

ID	Name, Location & Description	Card & Level
CSIP76	<p>File locator reference</p> <p>mets/fileSec/fileGrp/file/FLocat</p> <p>The location of each external file must be defined by the file location <FLocat> element using the same rules as references for 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.</p>	<p>1..1</p> <p>MUST</p>
CSIP77	<p>Type of locator</p> <p>mets/fileSec/fileGrp/file/FLocat[@LOCTYPE='URL']</p> <p>The locator type is always used with the value “URL” from the vocabulary in the attribute.</p>	<p>1..1</p> <p>MUST</p>
CSIP78	<p>Type of link</p> <p>mets/fileSec/fileGrp/file/FLocat[@xlink:type='simple']</p> <p>Attribute used with the value “simple”. Value list is maintained by the xlink standard.</p>	<p>1..1</p> <p>MUST</p>
CSIP79	<p>Resource location</p> <p>mets/fileSec/fileGrp/file/FLocat/@xlink:href</p> <p>The actual location of the resource. We recommend recording a URL type filepath within this attribute.</p>	<p>1..1</p> <p>MUST</p>
CSIP80	<p>Structural description of the package</p> <p>mets/structMap</p> <p>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. Every CSIP compliant METS file must include ONE structural map <structMap> element used exactly as described in this section of requirements. Institutions can add their own additional custom structural maps as separate <structMap> sections following their own requirements.</p>	<p>1..n</p> <p>MUST</p>
CSIP81	<p>Type of structural description</p> <p>mets/structMap[@TYPE='PHYSICAL']</p> <p>The mets/structMap/@TYPE attribute must take the value “PHYSICAL” from the vocabulary. See also: Structural map typing</p>	<p>1..1</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP82	<p>Name of the structural description mets/structMap[@LABEL='CSIP'] The <code>mets/structMap/@LABEL</code> attribute value is set to “CSIP” from the vocabulary. This requirement identifies the CSIP compliant structural map <code><structMap></code> element.See also: Structural map label</p>	1..1 MUST
CSIP83	<p>Structural description identifier mets/structMap[@LABEL='CSIP']/@ID An <code>xml:id</code> identifier for the structural description (structMap) used for internal package references. It must be unique within the package.</p>	1..1 MUST
CSIP84	<p>Main structural division mets/structMap[@LABEL='CSIP']/div The structural map comprises a single division.</p>	1..1 MUST
CSIP85	<p>Main structural division identifier mets/structMap[@LABEL='CSIP']/div/@ID Mandatory, <code>xml:id</code> identifier must be unique within the package.</p>	1..1 MUST
CSIP88	<p>Metadata division mets/structMap[@LABEL='CSIP']/div/div[@ LABEL='Metadata'] The metadata referenced in the administrative and/or descriptive metadata section is described in the structural map with one sub division. When the transfer consists of only administrative and/or descriptive metadata this is the only sub division that occurs.</p>	1..1 MUST
CSIP89	<p>Metadata division identifier mets/structMap[@LABEL='CSIP']/div/div[@ LABEL='Metadata']/@ID Mandatory, <code>xml:id</code> identifier must be unique within the package.</p>	1..1 MUST
CSIP90	<p>Metadata division label mets/structMap[@LABEL='CSIP']/div/div[@ LABEL='Metadata'] The metadata division <code><div></code> element's <code>@LABEL</code> attribute value must be “Metadata”.See also: File group names</p>	1..1 MUST

ID	Name, Location & Description	Card & Level
CSIP91	<p>Metadata division references administrative metadata</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Metadata']/@ADMID</p> <p>The administrative metadata division should reference all current administrative metadata sections. All <amdSec>s with @STATUS='CURRENT' SHOULD be referenced by their identifier, @ID. The current <amdSec> @IDs are recorded in the div[@LABEL='Metadata']/@ADMID attribute in a space delimited list.</p>	<p>0..1</p> <p>SHOULD</p>
CSIP92	<p>Metadata division references descriptive metadata</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Metadata']/@DMDID</p> <p>The descriptive metadata division should reference all current descriptive metadata sections. All <dmdSec>s with @STATUS='CURRENT' SHOULD be referenced by their identifier, @ID. The current <dmdSec> @IDs are recorded in the div[@LABEL='Metadata']/@DMDID attribute in a space delimited list.</p>	<p>0..1</p> <p>SHOULD</p>
CSIP93	<p>Documentation division</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']</p> <p>The documentation referenced in the file section file groups is described in the structural map with one sub division.</p>	<p>0..1</p> <p>SHOULD</p>
CSIP94	<p>Documentation division identifier</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']/@ID</p> <p>Mandatory, xml:id identifier must be unique within the package.</p>	<p>1..1</p> <p>MUST</p>
CSIP95	<p>Documentation division label</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']</p> <p>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</p>	<p>1..1</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP96	<p>Documentation file references</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']/fptr</p> <p>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 <fptr> element.</p>	<p>0..n</p> <p>MUST</p>
CSIP116	<p>Documentation file group reference pointer</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Documentation']/fptr/@FILEID</p> <p>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.</p>	<p>1..1</p> <p>MUST</p>
CSIP97	<p>Schema division</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']</p> <p>The schemas referenced in the file section file groups are described in the structural map within a single sub-division.</p>	<p>0..1</p> <p>SHOULD</p>
CSIP98	<p>Schema division identifier</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']/@ID</p> <p>Mandatory, <code>xml:id</code> identifier must be unique within the package.</p>	<p>1..1</p> <p>MUST</p>
CSIP99	<p>Schema division label</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']</p> <p>The schema division <div> element's @LABEL attribute has the value “Schemas” from the vocabulary. See also: File group names</p>	<p>1..1</p> <p>MUST</p>
CSIP100	<p>Schema file reference</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']/fptr</p> <p>All file groups containing schemas described in the package are referenced via the relevant file group identifiers. One file group reference per fptr-element</p>	<p>0..n</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP118	<p>Schema file group reference</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Schemas']/fptr/@FILEID</p> <p>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.</p>	1..1 MUST
CSIP101	<p>Content division</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']</p> <p>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.</p>	0..1 SHOULD
CSIP102	<p>Content division identifier</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']/@ID</p> <p>Mandatory, <code>xml:id</code> identifier must be unique within the package.</p>	1..1 MUST
CSIP103	<p>Content division label</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']</p> <p>The package’s content division <code><div></code> element must have the @LABEL attribute value “Representations”, taken from the vocabulary. See also: File group names</p>	1..1 MUST
CSIP104	<p>Content division file references</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']/fptr</p> <p>All file groups containing content described in the package are referenced via the relevant file group identifiers. One file group reference per fptr-element.</p>	0..n MUST
CSIP119	<p>Content division file group references</p> <p>mets/structMap[@LABEL='CSIP']/div/div[@LABEL='Representations']/fptr/@FILEID</p> <p>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.</p>	1..1 MUST

ID	Name, Location & Description	Card & Level
CSIP105	<p>Representation division</p> <p>mets/structMap[@LABEL='CSIP']/div/div</p> <p>When a package consists of multiple representations, each described by a representation level METS.xml document, there should be 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.</p>	<p>0..n</p> <p>SHOULD</p>
CSIP106	<p>Representations division identifier</p> <p>mets/structMap[@LABEL='CSIP']/div/div/@ID</p> <p>Mandatory, <code>xml:id</code> identifier must be unique within the package.</p>	<p>1..1</p> <p>MUST</p>
CSIP107	<p>Representations division label</p> <p>mets/structMap[@LABEL='CSIP']/div/div/@LABEL</p> <p>The package's representation division <code><div></code> element <code>@LABEL</code> attribute value must be the path to the representation level METS document starting with the value "Representations" followed by the main folder name for example "Representations/submission" and "Representations/ingest". This requirement gives the same value to be used as the requirement named "Description of the use of the file group" (CSIP64)See also: File group names</p>	<p>1..1</p> <p>MUST</p>
CSIP108	<p>Representations division file references</p> <p>mets/structMap[@LABEL='CSIP']/div/div/mptr/@xlink:title</p> <p>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.</p>	<p>1..1</p> <p>MUST</p>
CSIP109	<p>Representation METS pointer</p> <p>mets/structMap[@LABEL='CSIP']/div/div/mptr</p> <p>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.</p>	<p>1..1</p> <p>MUST</p>

ID	Name, Location & Description	Card & Level
CSIP110	Resource location mets/structMap/div/div/mptr/@xlink:href The actual location of the resource. We recommend recording a URL type filepath within this attribute.	1..1 MUST
CSIP111	Type of link mets/structMap/div/div/mptr[@xlink:type='simple'] Attribute used with the value “simple”. Value list is maintained by the xlink standard.	1..1 MUST
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
REF_METS_1	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_2	behaviorSec Section not defined or used in CSIP, additional own uses may occur. Information regarding the behaviour 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
Phillip Mike Tømmerholt	Danish National Archives
Kuldar Aas	Estonian National Archives
Koit Saarevet	Estonian National Archives
Jaime Kaminski	Highbury IVS
Miguel Ferreira	Keep Solutions

Name	Organisation
Helder Silva	Keep Solutions
Carl Wilson	Open Preservation Foundation
Karin Bredenberg	Kommunalförbundet Sydarkivera

I.I. Contributors to previous version

Name	Organisation
Karin Bredenberg	Kommunalförbundet Sydarkivera
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	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.

Revision	Date	Authors(s)	Organisation	Description
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
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 rising 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.

Revision	Date	Authors(s)	Organisation	Description
1.0	31.01.2017	Kuldar Aas	NAE	Final small editorial additions
1.1	14.05.2018	Kuldar Aas (editor), DILCIS Board DI	NAE LCIS Board	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.
2.0-DFT	28.11.2018	DILCIS Board, E-ARK4ALL	DILCIS Board	Specification updated and released for open review.
2.0.0	31.05.2019	DILCIS Board E-ARK4ALL	DILCIS Board	Specification updated after open review.
2.0.1	09.09.2019	Karin Bredenberg, C. Wilson, & H. Silva	NAS, OPF KEEPS	Correction for @LABEL and @USE attributes, typos, layout and PDF formatting.
2.0.2	28.10.2019	Karin Bredenberg	SYD	Fixed schema paths
2.0.3	08.01.2020	Karin Bredenberg, & C. Wilson	SYD OPF	Fixed error in value list note type.
2.0.4	12.06.2020	K.Bredenberg, C. Wilson, & J. Kaminski	SYD OPF HIGH	Update in example ID:s, preface text and output display update
2.1.0	15.10.2021	K.Bredenberg & C. Wilson	SYD OPF	Specification updated to version 2.1. Language updates and clarifications to requirements. Updated list of approved CITS specifications. Removed redundant requirement.

III. Acknowledgements

The Common Specification for Information Packages was first developed within the E-ARK project in 2014 – 2017. E-ARK was an EC-funded pilot action project in the Competiveness and Innovation Programme 2007-2013, Grant Agreement no. 620998 under the Policy Support Programme.

We would like to thank the National Archives of Sweden and Karin Bredenberg for their support and the availability of the Swedish national Common Specifications, upon which most of this document has been built.

The authors of this deliverable would like to thank all national archives, tool developers and other stakeholders

who provided valuable knowledge about their requirements for information packages and feedback to this specification!

IV. Contact & Feedback

The Common Specification for Information Packages is maintained by the Digital Information LifeCycle Interoperability Standard Board (DILCIS Board). For further information about the DILCIS Board or feedback on the current document please consult the website <http://www.dilcis.eu/> or contact us at <mailto:info@dilcis.eu>.