

---

# Common Specification for Preservation Metadata

E-ARK Common Specification for Preservation Metadata  
using PREMIS

Date: 17.05.2024

Version: 1.0.1

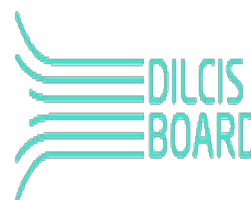
# Common Specification for Preservation Metadata

E-ARK Common Specification for Preservation Metadata using PREMIS



The European Commission eArchiving procurement recognizes the E-ARK specifications as the eArchiving specifications which are funded under the eArchiving Common Services Platform Agreement No. LC-01905904-CNECT/LUX/2021/OP/0077.

This specification is published, supported, and developed by the Digital Information LifeCycle Interoperability Standards (DILCIS) Board under the auspices of the DLM Forum.



This specification is maintained by Digital Information LifeCycle Interoperability Standards Board and is licensed under CC BY 4.0



This specification was previously developed with the support of the European Union:

E-ARK Grant No: 620998 CIP-ICT-PSP.2013.2.5  
E-ARK4ALL Agreement No. LC-00921441 CEF-TC-2018-15  
E-ARK3 Agreement No. LC-01390244 CEF-TC-2019-3

# 1 Preface

## 1.1 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:

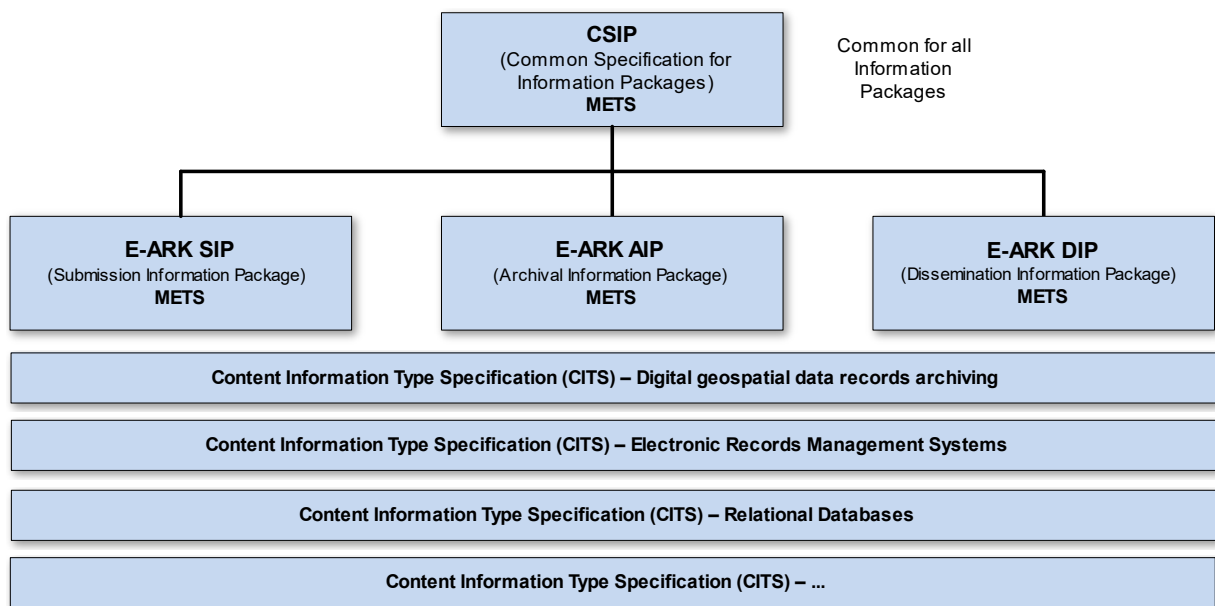


Figure I: Diagram showing E-ARK specification dependency hierarchy. Note that the image only shows a selection of the published CITS and isn't an exhaustive list.

Specification	Aim and Goals
<b>Common Specification for Information Packages</b>	<p>This document introduces the concept of a Common Specification for Information Packages (CSIP). Its three main purposes are to:</p> <ul style="list-style-type: none"> <li>Establish a common understanding of the requirements, which need to be met in order to achieve interoperability of Information Packages.</li> <li>Establish a common base for the development of more specific Information Package definitions and tools within the digital preservation community.</li> </ul>

Specification	Aim and Goals
	<ul style="list-style-type: none"> <li>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.</li> </ul> <p>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.</p>
<b>E-ARK SIP</b>	<p>The main aims of this specification are to:</p> <ul style="list-style-type: none"> <li>Define a general structure for a Submission Information Package format suitable for a wide variety of archival scenarios, e.g. document and image collections, databases or geographical data.</li> <li>Enhance interoperability between Producers and Archives.</li> <li>Recommend best practices regarding metadata, content and structure of Submission Information Packages.</li> </ul>
<b>E-ARK AIP</b>	<p>The main aims of this specification are to:</p> <ul style="list-style-type: none"> <li>Define a generic structure of the AIP format suitable for a wide variety of data types, such as document and image collections, archival records, databases or geographical data.</li> <li>Recommend a set of metadata related to the structural and the preservation aspects of the AIP as implemented by the eArchiving Reference Implementation (earkweb).</li> <li>Ensure the format is suitable to store large quantities of data.</li> </ul>
<b>E-ARK DIP</b>	<p>The main aims of this specification are to:</p> <ul style="list-style-type: none"> <li>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.</li> <li>Recommend a set of metadata related to the structural and access aspects of the DIP.</li> </ul>
<b>Content Information Type Specifications</b>	<p>The main aim and goal of a Content Information Type Specification is to:</p> <ul style="list-style-type: none"> <li>Define, in technical terms, how data and metadata must be formatted and placed within a CSIP Information Package in order to achieve interoperability in exchanging specific Content Information.</li> </ul> <p>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, <a href="http://dilcis.eu/">http://dilcis.eu/</a>).</p>

## 1.2 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://eak-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/>.

### **1.3 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

<b>1</b>	<b>Preface</b> .....	<b>3</b>
<b>1</b>	<b>Context</b> .....	<b>9</b>
1.1	Purpose.....	9
1.2	Scope .....	9
<b>2</b>	<b>Preservation metadata</b> .....	<b>9</b>
2.1	PREMIS .....	9
2.2	Use of PREMIS in the E-ARK specifications supporting information packages .....	10
2.2.1	The version of PREMIS to be used with this specification.....	10
2.2.2	The conformance of the specification towards PREMIS .....	10
2.2.3	PREMIS XML-documents .....	10
2.2.4	Vocabularies .....	12
2.2.5	A note on identifiers .....	12
<b>3</b>	<b>PREMIS data model</b> .....	<b>12</b>
3.1	PREMIS fundamentals .....	13
<b>4</b>	<b>A focus on XML encoding of PREMIS in this specification</b> .....	<b>14</b>
4.1	PREMIS object .....	14
4.1.1	Object identifier.....	15
4.1.2	Checksums .....	15
4.1.3	File format .....	15
4.1.4	Object characterisation .....	16
4.1.5	Original name .....	16
4.1.6	Storage.....	16
4.1.7	Relationship .....	17
4.1.8	Linking rights statement .....	17
4.2	PREMIS event .....	18
4.2.1	Event identifier .....	18
4.2.2	Link to agent/object.....	18
4.2.3	Migration event type .....	18
4.3	PREMIS agent .....	20
4.4	PREMIS rights .....	20
4.5	Metadata regarding Files, Representations and Creating Software in a SIP .....	21
4.5.1	Identifying the file which has been created and the software for creation.....	21
4.5.2	Identifying a representation .....	22
4.5.2.1	Identifying the software used in creation of the representation .....	22

---

4.5.2.2	Establishing the relationship between a representation and creation software.....	23
4.5.3	Metadata regarding Representations and Access Software in a DIP .....	24
4.5.3.1	Identifying the representation to be rendered.....	25
4.5.3.2	Identifying the software for rendering .....	26
4.5.3.3	Establishing the relationship between a representation and rendering software .....	26
<b>5</b>	<b>Metadata in the Information Package.....</b>	<b>27</b>
5.1	Using the Common Specification for Preservation Metadata together with the Common Specification for Information Packages (CSIP) .....	28
5.1.1	Specific fields to use in CSIP for the preservation metadata.....	28
5.2	Placement of preservation metadata documents in a CSIP Information Package .....	28
<b>6</b>	<b>Metadata.....</b>	<b>28</b>
6.1	Use of PREMIS .....	28
6.1.1	Use of the PREMIS root element (element premis) .....	29
6.1.1.1	Location of XML schema.....	29
6.1.2	Metadata for an object of entity type “Intellectual Entity” and specifically of type “Environment” .....	30
6.1.3	Metadata for an object of entity type “Representation” .....	32
6.1.4	Metadata for object of entity type “File” .....	34
6.1.5	Metadata for an agent.....	41
6.1.6	Metadata for an event.....	42
6.1.7	Metadata for rights.....	44
6.2	Examples.....	49
6.2.1	Example of representation and its access software.....	49
6.2.2	Example of a file with an associated agent, event and rights .....	51
<b>7</b>	<b>Postface.....</b>	<b>55</b>

---

## TABLE OF FIGURES

Figure 1: Image showing the placement of a single PREMIS document .....	11
Figure 2: The placement of multiple PREMIS documents .....	12
Figure 3: PREMIS data model (with permission from the PREMIS Editorial Committee) .....	13

## LIST OF TABLES

Table 1: Specific fields to use in CSIP for a PREMIS document .....	28
Table 2: PREMIS root element.....	30
Table 3: Description of the software for rendering.....	30
Table 4: Description of the data object for creation or rendering.....	32
Table 5: Description of the data object of entity type file .....	34
Table 6: Description of the agent .....	41
Table 7: Description of the event.....	42
Table 8: Description of the rights .....	44



## 1 Context

### 1.1 Purpose

The purpose of this document is to describe the Common Specification for Preservation Metadata for digital objects based on the use of the de-facto standard Preservation Metadata: Implementation Strategies (PREMIS) (<http://www.loc.gov/standards/premis/>). The document is a simple and basic implementation of PREMIS, and a more elaborate implementation can and should be made in your preservation platform. We encourage extending this specification in your implementation in your preservation system with more requirements, even if those introduced here are a sound foundation.

The purpose of this specification is to maintain the authenticity and integrity of digital objects in digital storage. It is not for the discovery of the digital objects in any other sense than for the preservation of the objects. The specification is designed to be used for the transfer to archives and information exchange between different systems requiring preservation metadata for the digital objects of the transfer. This specification is supported by an XML schema and by an ontology.

### 1.2 Scope

The scope of this specification is to provide the functionality and maintain the authenticity and integrity of the digital objects in an information package by using preservation metadata. The specification only outlines the use of the PREMIS XML schema, even if it is possible to implement PREMIS with an ontology or other preferred way.

## 2 Preservation metadata

Preservation metadata is the information a preservation repository uses to support the digital preservation process – where a preservation repository is a repository that undertakes the long-term preservation of the digital objects in its custody, either as its sole responsibility or as one of many responsibilities. For preservation metadata, the standard used is Preservation Metadata: Implementation Strategies (PREMIS).

### 2.1 PREMIS

PREMIS (Preservation Metadata: Implementation Strategies) and its associated PREMIS Data Dictionary is a comprehensive, practical resource for implementing preservation metadata in digital preservation systems. The Data Dictionary defines preservation metadata that:

- Supports the viability, renderability, understandability, authenticity and identity of digital objects in a preservation context;
- Represents the information most preservation repositories need to know to preserve digital materials over the long term;
- Emphasises “implementable metadata”: rigorously defined, supported by guidelines for creation, management, and use, and is oriented toward automated workflows; and,
- Embodies technical neutrality: no assumptions are made about preservation technologies, strategies, metadata storage and management, etc.

---

## 2.2 Use of PREMIS in the E-ARK specifications supporting information packages

When using preservation metadata together with the Common Specification for Information Packages (CSIP) (<http://earkcsip.dilcis.eu>), it is recommended that these are included in the information package in PREMIS format. Although this is not mandatory, all tools claiming to be able to validate CSIP compliant Information Packages must also be able to validate PREMIS metadata once it exists within the package. The two high-level requirements for the use of PREMIS in Common Specification IPs are that:

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

Further, to enhance the interoperability of the CSIP and to strengthen the management of information packages (IPs) in an archive, this specification imposes additional requirements regarding the use of PREMIS for describing IPs. The principles adopted in the CSIP for deciding which additional PREMIS semantic units are required are:

- PREMIS should be used to record detailed technical metadata;
- Technical information should be included in PREMIS metadata by using extension schemas;
- Information about agents carrying out preservation actions must be recorded in the PREMIS metadata (this is because the METS agents describe agents relevant for generic IP level events, such as the creation or submission of the package, not the preservation of the data);
- Event descriptions should be included in PREMIS metadata. Use of the official PREMIS event vocabulary (<https://id.loc.gov/vocabulary/preservation/eventType.html>) is recommended (note that more elaborate descriptions can be made than are made in this specification);
- Detailed rights information should be included in PREMIS. High-level rights information in METS indicates restrictions. Detailed, object-specific rights information will be included in the PREMIS metadata;
- File format information for all files should be included as Persistent Unique Identifier (PUIID) values in the appropriate PREMIS semantic units.

### 2.2.1 The version of PREMIS to be used with this specification

Version 3.0 of the PREMIS data dictionary is used in this specification (<http://www.loc.gov/standards/premis/v3/index.html>).

### 2.2.2 The conformance of the specification towards PREMIS

The PREMIS standard defines a conformance scheme (<https://www.loc.gov/standards/premis/premis-conformance-20150429.pdf>); this specification is conformant following level 1B conformance through mapping of the entities object, event and agent.

### 2.2.3 PREMIS XML-documents

It is strongly recommended that PREMIS metadata is kept in discrete PREMIS XML files inside the IP. The PREMIS metadata can be included in the IP in separate files, and this specification does not implement a convention regarding the naming and numbering of the PREMIS files. For all the different information packages, the demands on naming will differ and depend upon the receiving body where the recommendation is to follow the CSIP and use the name PREMIS.xml. This needs to be stated in the

submission agreement for the Submission Information Package (SIP), and the placement in the IP is shown in Figure 1. Naming in the preservation system responsible for the Archival Information Package (AIP) will be vendor dependent and have its own implementation. In the case of the Dissemination Information Package (DIP) being delivered to a consumer, the recommendation is to name the PREMIS XML document following the Common Specification for Information Packages (CSIP, <https://dilcis.eu/specifications/common-specification>) and name it PREMIS.xml.

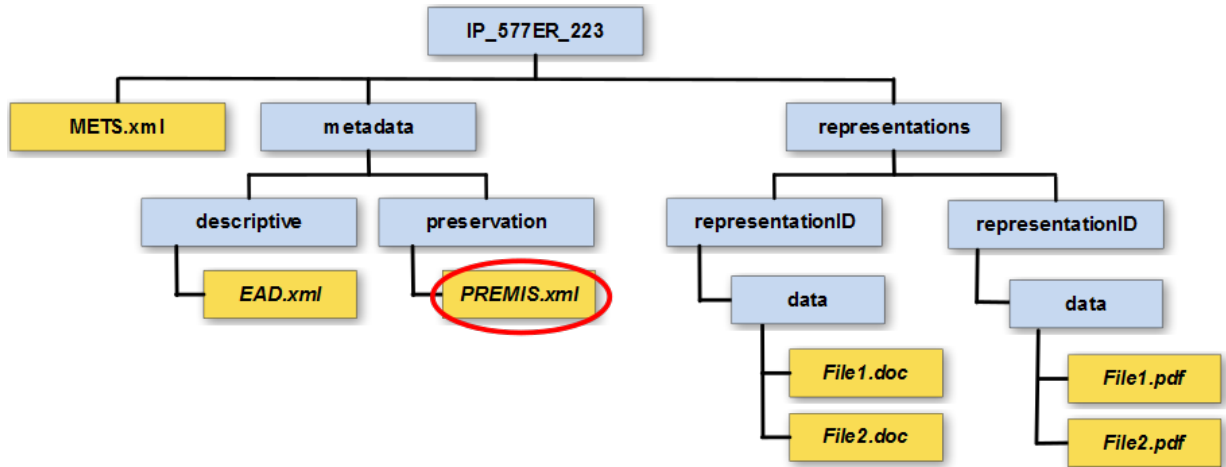
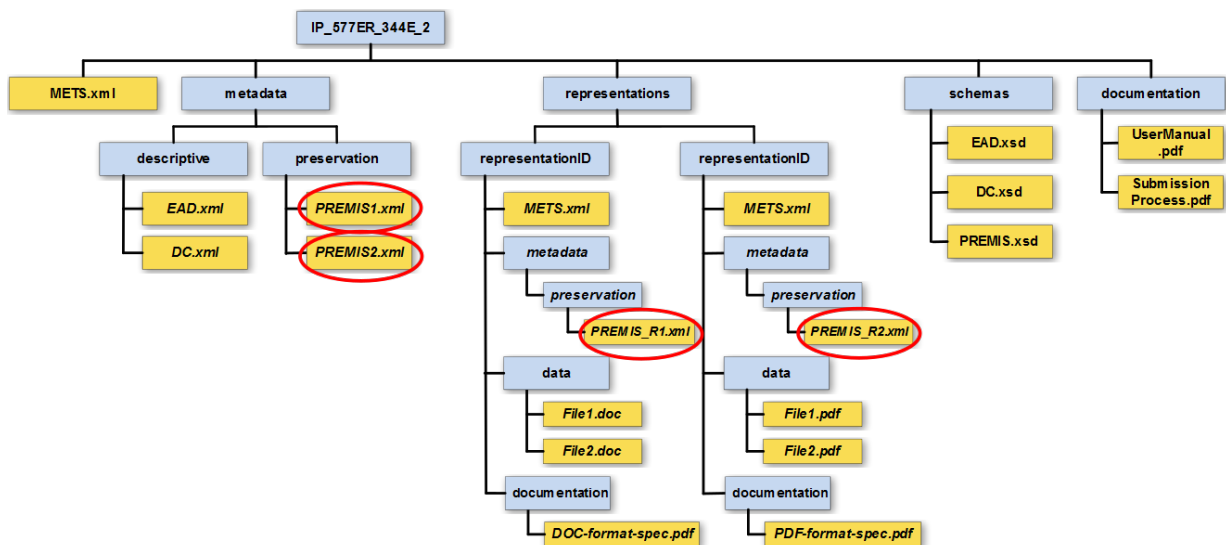


Figure 1: Image showing the placement of a single PREMIS document

Implementations can either store all preservation metadata in a single PREMIS file or split them into multiple files. In this case, the only requirement is that all PREMIS files must be listed in the appropriate METS file shown in Figure 2 (i.e. package PREMIS files from the package METS file and representation PREMIS files from the representation METS files, and referenced in the METS file(s) using the mdRef attributes and elements at the appropriate places described in the CSIP specification). This follows the recommendations from the PREMIS Editorial Committee and the METS Board (<http://www.loc.gov/standards/premis/guidelines2017-premismets.pdf>).



---

**Figure 2: The placement of multiple PREMIS documents**

### 2.2.4 Vocabularies

This specification does not present any new definitions or a definitive list of vocabularies for use with PREMIS semantic units. Instead, it recommends the use of the Library of Congress hosted vocabularies developed specifically to provide values for various PREMIS semantic units. All relevant vocabularies are presented in the PREMIS Data Dictionary together with the semantic unit it is used in. All the available vocabularies can be found at this web resource: <https://id.loc.gov/vocabulary/preservation.html>.

### 2.2.5 A note on identifiers

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

- [entity]IdentifierType
- [entity]IdentifierValue

The PREMIS data dictionary and this specification recognises that the use of identifier types is an implementation-specific issue and does not recommend or require vocabularies for identifier types. The Library of Congress has a Standard Identifiers Scheme (<https://id.loc.gov/vocabulary/identifiers.html>). Its use is recommended in this specification instead of implementation-specific vocabularies.

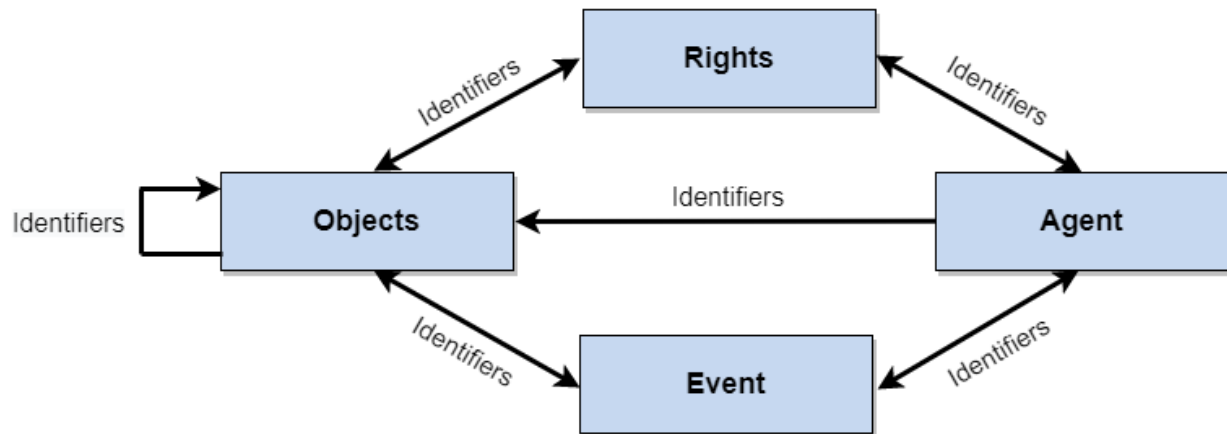
This specification does not mandate how the identifier values are set up in the system. There are several different ways to create a long-term stable identifier value, including using a prefix to identify the identifier type also in the value and adding the version of a software that is being used in the identifier value. It is also important to remember that there needs to be possible to connect the PREMIS information with the correct file in the information package, thus making sure that the identifiers or the filename are possible to cross-link.

How the values are set up in the transmission moment needs to be agreed upon in the submission agreement, especially in the cases when the SIP will include preservation metadata.

All identifiers used throughout this specification are just examples, and for ease of reading and understanding in the examples, the local identifier type has been defined as “filepath”, and the identifier value is a file path to the digital object.

## 3 PREMIS data model

The PREMIS Data Dictionary defines semantic units. Each semantic unit defined in the Data Dictionary is mapped to an entity that is organised within a simple data model. A semantic unit can, therefore, be understood as a property of an entity. The model defines four entities important regarding digital preservation activities: Objects, Events, Agents and Rights. Figure 3 provides a graphical illustration of the PREMIS Data Model.



**Figure 3: PREMIS data model (with permission from the PREMIS Editorial Committee)**

In Figure 3, entities are represented by boxes and relationships between entities are represented by arrows. When arrows are bi-directional, then each entity type contains a semantic unit allowing it to link to the other. So, for example, the Rights entity includes a semantic unit recording information about the relationship with an Agent, and the Agent entity includes a semantic unit recording information about associated Rights.

The entities in the PREMIS data model are defined as follows:

- **Object (or Digital Object):** A discrete unit of information subject to digital preservation. Version 3 introduces the notion that this can be an environment used as part of the preservation process.
- **Environment:** Technology (software or hardware) supporting a Digital Object in some way (e.g. rendering or execution). Environments can be described as Intellectual Entities and captured and preserved in the preservation repository as Representations, Files and/or Bitstreams.
- **Event:** An action that involves or affects at least one Object or Agent associated with or known by the preservation repository.
- **Agent:** A person, organisation, or software program/system associated with Events in the life of an Object, or with Rights attached to an Object. It can also be related to an environment Object that acts as an Agent.
- **Rights Statement:** Assertion of one or more Rights or permissions pertaining to an Object and/or Agent.

In order to thoroughly understand PREMIS, it is recommended that users study the Data Dictionary and participate in the events led by the PREMIS Editorial Committee. More information can be found on the PREMIS website.

### 3.1 PREMIS fundamentals

The standard describes all its elements in the Data Dictionary available online at <http://www.loc.gov/standards/premis/v3/index.html>.

---

Elements are called semantic components, and each semantic component has its own entry in the Data Dictionary. A semantic unit that has semantic components does not have any value of its own. Only semantic units at the lowest level have values. Thus:

- **Semantic Component**, which is a grouping of semantic units and,
- **Semantic Units**, which are the lowest level of description.

It is important to remember that PREMIS is implementation-independent, but the XML schema has been chosen as the implementation form in this specification. Thus, it is possible to implement these structures in, for example, a database and export it as the XML document when a transfer of the information with its preservation metadata is performed.

## 4 A focus on XML encoding of PREMIS in this specification

As previously mentioned, PREMIS is used to describe technical metadata of digital objects, rights metadata to define the rights status in relation to specific agents or for specific objects and to record events that are relevant regarding the digital provenance of digital objects.

It is possible to use this specification for all the information packages in the OAIS Reference Model. When the information package is stored in an OAIS Reference Model-based archive, the basic information needed is shown in this specification. If the software for storing information packages is purchased from a vendor, this will already be covered in their design, but every repository, especially if the work consists of designing its own e-archive, should perform a preservation planning exercise and thoroughly investigate PREMIS and extend this PREMIS profile according to their own requirements.

Observe that in the following sections, and examples code snippets are shown, which may not contain all requirements found in the requirements tables.

### 4.1 PREMIS object

The PREMIS object contains technical information about a digital object. The digital object can be of different entity types:

- **Intellectual Entity**: A set of content that is considered a single intellectual unit for purposes of management and description (for example, a particular book, map, photograph, database, or piece of hardware or software). An Intellectual Entity can include other Intellectual Entities (for example, a website can include a web page; a web page can include an image). An Intellectual Entity may have one or more digital representations.
- **Representation**: A digital or physical object embodying an Intellectual Entity. A digital representation is the set of stored digital files and structural metadata needed to provide a complete and reasonable rendition of the Intellectual Entity. A physical representation is an item such as a manuscript, video cassette, or printed document.
- **File**: A named and ordered sequence of bytes that is known to an operating system.
- **Bitstream**: Contiguous or non-contiguous data within a file that has meaningful properties for preservation purposes.

**Note:** In this specification, the use of all entity types are described except bitstream. While it is possible to use the entity type bitstream, it is not recommended.

#### 4.1.1 Object identifier

**PREMIS-ID-LOCAL:** If an identifier of type local is used, this identifier SHOULD be unique in the scope of the PREMIS document as well as in the repository.

**PREMIS-ID-OTHER:** Other object identifiers of the allowed types MAY be used additionally to the identifier of type local by repeating the whole “objectIdentifier” element.

The example below shows three examples, one of an identifier of type local, one of the type UUID and lastly, the example of the local identifier type file path, which is being used throughout this specification to make the examples easier to read.

```
<objectIdentifier>
  <objectIdentifierType>local</objectIdentifierType>
  <objectIdentifierValue>fileId001</objectIdentifierValue>
</objectIdentifier>
<objectIdentifier>
  <objectIdentifierType>UUID</objectIdentifierType>
  <objectIdentifierValue>5707ecac-6713-41b5-826a-3388e21c4c2f
  </objectIdentifierValue>
</objectIdentifier>
<objectIdentifier>
  <objectIdentifierType>filepath</objectIdentifierType>
  <objectIdentifierValue>metadata/file.xml</objectIdentifierValue>
</objectIdentifier>
```

#### 4.1.2 Checksums

**PREMIS-CHECKSUMS:** Checksums SHOULD be provided as a descendant of the “objectCharacteristics” element information in the form of the recommended SHA-256 hashsum, a fixed size 256-bit value. An example of a value shortened hashsum is shown below.

```
<fixity>
  <messageDigestAlgorithm>SHA-256</messageDigestAlgorithm>
  <messageDigest>3b1d00f7871d9102001c77f...</messageDigest>
  <messageDigestOriginator>/usr/bin/sha256sum</messageDigestOriginator>
</fixity>
```

#### 4.1.3 File format

**PREMIS-FILE-FORMAT:** The format element SHOULD be provided either using the “formatRegistry” or the “formatDesignation” element subelements, or both.

**PREMIS-FILE-FORMAT-PUID:** Regarding the formatRegistry, the Persistent Unique Identifier (PUID) based on the PRONOM technical registry SHOULD be used. An example is shown below.

```
<format>
  <formatDesignation>
    <formatName>XML</formatName>
    <formatVersion>1.0</formatVersion>
  </formatDesignation>
  <formatRegistry>
    <formatRegistryName>PRONOM</formatRegistryName>
    <formatRegistryKey>fmt/101</formatRegistryKey>
    <formatRegistryRole>specification</formatRegistryRole>
  </formatRegistry>
</format>
```

#### 4.1.4 Object characterisation

**PREMIS-CHARACTERISATION:** The JHOVE14 technical characterisation result (XML format) MAY be embedded as a descendant of the objectCharacteristicsExtension element. An example is shown below.

```
<objectCharacteristicsExtension>
  <jhove>
    ...
  </jhove>
</objectCharacteristicsExtension>
```

#### 4.1.5 Original name

**PREMIS-ORIGINAL-NAME:** The originalName element MAY be used to record the name of an original file. An example is shown below.

```
<originalName>originalfilename.ext</originalName>
```

#### 4.1.6 Storage

**PREMIS-STORAGE:** The storage element MAY hold information about the physical location of the digital object and will most likely only be present in the storage system, not in the transfer of the SIP.

Ideally, this is a resolvable Uniform Resource Identifier (URI), but it can also generally hold information needed to retrieve the digital object from the storage system (e.g. access control or for segmented IPs). An example is shown below.

```
<storage>
  <contentLocation>
    <contentLocationType>Uniform Resource Identifier</contentLocationType>
    <contentLocationValue>
      /path/to/file.txt
    </contentLocationValue>
  </contentLocation>
</storage>
```



---

```
</contentLocation>
  <storageMedium>hard disk HD2253</storageMedium>
</storage>
```

#### 4.1.7 Relationship

**PREMIS-RELATIONSHIP:** The relationship element SHOULD be used to describe relationships of the digital object.

**PREMIS-IP-INCLUDED:** If an IP is part of another IP, then the element relationshipSubType MUST reference the superordinated IP. An example of the latter case is shown below.

```
<relationship>
  <relationshipType>structural</relationshipType>
  <relationshipSubType>is included in</relationshipSubType>
  <relatedObjectIdentifier>
    <relatedObjectIdentifierType>local</relatedObjectIdentifierType>
    <relatedObjectIdentifierValue>
      ID123e4567-e89b-12d3-a456-426655440000
    </relatedObjectIdentifierValue>
  </relatedObjectIdentifier>
</relationship>
```

Please observe that there is also referencing made between the different parts of an IP and to other representations in the METS document. This referencing MUST be created following the description in the Common Specification for Information Packages (CSIP).

#### 4.1.8 Linking rights statement

**PREMIS-RIGHTS:** The linkingRightsStatementIdentifier element MAY be used to describe rights statements attached to the object.

For example, only files that have the “discovery right” are being indexed in order to allow these files to be retrievable by the full-text search. An example of the latter case is shown below.

```
<linkingRightsStatementIdentifier>
  <linkingRightsStatementIdentifierType>
    filepath
  </linkingRightsStatementIdentifierType>
  <linkingRightsStatementIdentifierValue>
    metadata/file.xml
  </linkingRightsStatementIdentifierValue>
</linkingRightsStatementIdentifier>
```

---

## 4.2 PREMIS event

### 4.2.1 Event identifier

**PREMIS-EVENT-ID:** The eventIdentifier SHOULD be used to identify events, such as preservation actions, which were applied. An example of the identifier is shown below.

Observe that there are more information given in the event, which can be found in table [event]

```
<eventIdentifier>
  <eventIdentifierType>local</eventIdentifierType>
  <eventIdentifierValue>PDFtoPDF/A</eventIdentifierValue>
</eventIdentifier>
```

### 4.2.2 Link to agent/object

**PREMIS-EVENT-AGENT:** If an event is described, the agent which caused the event (e.g. person, software, hardware, etc.). MUST be related to the event by means of the linkingAgentIdentifier element.

In the example shown below the SIP to AIP conversion software is linked as an agent with identifier value "Sip2Aip", and the corresponding object is linked by the local type file path value.

```
<linkingAgentIdentifier>
  <linkingAgentIdentifierType>local</linkingAgentIdentifierType>
  <linkingAgentIdentifierValue>Sip2Aip</linkingAgentIdentifierValue>
</linkingAgentIdentifier>
<linkingObjectIdentifier>
  <linkingObjectIdentifierType>filepath</linkingObjectIdentifierType>
  <linkingObjectIdentifierValue>metadata/file.xml</linkingObjectIdentifierValue>
</linkingObjectIdentifier>
```

### 4.2.3 Migration event type

**PREMIS-EVENT-AGENT:** The event by which a resource was created SHOULD be recorded by means of the relatedEventIdentifier element in the object itself.

In the example below, the migration event is shown. The event expresses the fact that the object metadata/file.xml is the result of the migration event "migration-001", and the event which created the source object is "ingest-001".

```
<event>
  <eventIdentifier>
    <eventIdentifierType>local</eventIdentifierType>
    <eventIdentifierValue>migration-001</eventIdentifierValue>
  </eventIdentifier>
  <eventType>MIGRATION</eventType>
  <eventDateTime>2015-09-01T01:00:00+01:00</eventDateTime>
```

---

```

<eventOutcomeInformation>
  <eventOutcome>success</eventOutcome>
</eventOutcomeInformation>
<linkingAgentIdentifier>
  <linkingAgentIdentifierType>local</linkingAgentIdentifierType>
  <linkingAgentIdentifierValue>
    FileFormatConversion001
  </linkingAgentIdentifierValue>
</linkingAgentIdentifier>
<linkingObjectIdentifier>
  <linkingObjectIdentifierType>filepath</linkingObjectIdentifierType>
  <linkingObjectIdentifierValue>
    metadata/file.xml
  </linkingObjectIdentifierValue>
</linkingObjectIdentifier>
</event>

```

The outcome of the event is shown in the example below, and the event which created the source object is “ingest-001”.

Observe that it is possible to elaborate and extend the number of linking objects to also include the source object, which is important in the case of transforming a SIP to an AIP.

```

<object xsi:type='file'>
  <objectIdentifier >
    <objectIdentifierType>filepath</linkingObjectIdentifierType>
    <objectIdentifierValue>
      metadata/file.xml
    </objectIdentifierValue>
  </objectIdentifier>
<!-- PREMIS file continues but elements left out in this listing-->
<relationship>
  <relationshipType>derivation</relationshipType>
  <relationshipSubType>documents</relationshipSubType>
  <relatedObjectIdentifier>
    <relatedObjectIdentifierType>
      local
    </relatedObjectIdentifierType>
    <relatedObjectIdentifierValue>
      ID123e4567-e89b-12d3-a456-426655440000
    </relatedObjectIdentifierValue>
  </relatedObjectIdentifier>
  <relatedEventIdentifier>
    <relatedEventIdentifierType>local</relatedEventIdentifierType>
    <relatedEventIdentifierValue>
      ingest-001
    </relatedEventIdentifierValue>

```

---

```

    </relatedEventIdentifier>
  </relationship>
</object>

```

### 4.3 PREMIS agent

**PREMIS-AGENT:** Agents that are referenced in events **MUST** be described by means of the agent element. The software for indexing named “Indexing Software”, which supports full-text search of the items contained in a package, is shown below. In this case, the “discovery right” is assigned to this agent. Observe that the agent can carry more information than shown in this example.

```

<agent>
  <agentIdentifier>
    <agentIdentifierType>local</agentIdentifierType>
    <agentIdentifierValue>Indexer</agentIdentifierValue>
  </agentIdentifier>
  <agentName>Indexing Software</agentName>
  <agentType>Software</agentType>
  <linkingRightsStatementIdentifier>
    <linkingRightsStatementIdentifierType>
      local
    </linkingRightsStatementIdentifierType>
    <linkingRightsStatementIdentifierValue>
      discovery-right-001
    </linkingRightsStatementIdentifierValue>
  </linkingRightsStatementIdentifier>
</agent>

```

### 4.4 PREMIS rights

**PREMIS-RIGHTS:** Rights which are referenced in rights statements **MUST** be described by means of the rights element. The rights element holds information about the rights status of individual digital objects or about agents. An example is shown below.

```

<rights>
  <rightsStatement>
    <rightsStatementIdentifier>
      <rightsStatementIdentifierType>
        local
      </rightsStatementIdentifierType>
      <rightsStatementIdentifierValue>
        discovery-right-001
      </rightsStatementIdentifierValue>
    </rightsStatementIdentifier>
    <rightsBasis>Statute</rightsBasis>
    <rightsGranted>
      <act>displaying</act>
    </rightsGranted>
  </rightsStatement>
</rights>

```

---

```

    </rightsGranted>
  </rightsStatement>
</rights>

```

#### 4.5 Metadata regarding Files, Representations and Creating Software in a SIP

For preservation purposes, the creating application of a file or a representation can be critical in guaranteeing the authenticity of the digital objects. The possibility to add this information to the SIP in the transfer depends upon numerous factors, such as how the application for creating the package is created, to which information can be automatically extracted from the digital objects themselves. Therefore, this is seen as a strong recommendation for needed metadata in the transfer, but the specification is fully aware that the possibility to add the information might not exist.

In PREMIS, a representation is a “set of files, including structural metadata, needed for a complete and reasonable rendition of an Intellectual Entity.” (see page 8 of the Data Dictionary; <http://www.loc.gov/standards/premis/v3/premis-3-0-final.pdf>).

One of the core concepts in PREMIS is the definition of a representation, but it is also important to note that when using the CSIP structure, there is also an incorporation of physical management of different representations. When implementing PREMIS in CSIPs and especially in a SIP, it is necessary to decide if PREMIS files exist at representation level or at root level only (see more in the CSIP) and how fine-grained each description should be.

Three pieces of information are needed to ensure the knowledge of the representation which has been created and transferred as the SIP, which is expressed with the help of PREMIS:

- The first identifies the representation which has been created;
- The second identifies the software which was used in the creation;
- The third establishes the relationship between what is created and the software performing the creation.

At the same time, in the transfer of the digital objects, it is possible to give a description on file level and thus describe the creation of each file transferred can be made.

##### 4.5.1 Identifying the file which has been created and the software for creation

It is possible to describe each file being created and transferred with its connected software. See the example below which MUST be followed to describe the file and its creating software:

```

<object xsi:type='file'>
  <objectIdentifier>
    <objectIdentifierType>filepath</linkingObjectIdentifierType>
    <objectIdentifierValue>
      representations/data/image.tiff
    </objectIdentifierValue>
  </objectIdentifier>
<!-- PREMIS file continues but elements left out in this listing-->
  <objectCharacteristics>
    <creatingApplication>
      <creatingApplicationName>

```

---

```

        GIMP
        </creatingApplicationName>
        <creatingApplicationVersion>
            2.10.12
        </creatingApplicationVersion>
        <dateCreatedByApplication>
            2020-01-13
        </ dateCreatedByApplication >
    </creatingApplication>
    <objectCharacteristics>
<!-- PREMIS file continues but elements left out in this listing-->
</object>

```

#### 4.5.2 Identifying a representation

In order to describe the specific representation format in a PREMIS available semantic component (a grouping unit) for objects named “significantProperties” MUST be used. An example is shown below. In this case, the representation object is placed in the representations subfolder for the specific representation in the package in the subfolder named AVID.SA.18006.rep0 where all the digital objects significant property is that being created in the format SIARD2.

```

<object xsi:type="representation">
    <objectIdentifier>
        <objectIdentifierType>filepath</objectIdentifierType>
    <objectIdentifierValue>
        representations\AVID.SA.18006.rep0\
    </objectIdentifierValue>
</objectIdentifier>
    <significantProperties>
        <significantPropertiesType>
            Created format
        </significantPropertiesType>
        <significantPropertiesValue>SIARD2</significantPropertiesValue>
    </significantProperties>
<!-- PREMIS file continues but further elements left out in this listing-->
</object>

```

Note that the object type is “representation” and that the objectIdentifierType value is using for example purposes the type “filepath”, which is local to the IP scope value. The objectIdentifierValue should be using a UUID, for example.

##### 4.5.2.1 Identifying the software used in creation of the representation

With the release of PREMIS 3.0 a description of an environment is seen as an intellectual entity type object, which means that both non-environmental objects and environmental objects exist and can be described with PREMIS. The creation software is, therefore, an environmental object which per default is an intellectual entity following the definitions in PREMIS. In PREMIS a semantic unit named “environmentFunction” is available for objects which are conceived to describe the environment

object(s) functions with different levels of granularity. It is suggested to use the vocabulary available and recommended in PREMIS for describing the environment function <https://id.loc.gov/vocabulary/preservation/environmentFunctionType.html>. When describing an environment, it is recommended that the semantic unit “environmentDesignation” is used for information identifying the environment by using human-readable language, which can be expected to be understood outside of a digital repository.

See the example below which MUST be followed to describe the software following this vocabulary:

```
<object xsi:type="intellectualEntity">
  <objectIdentifier>
    <objectIdentifierType>local</objectIdentifierType>
    <objectIdentifierValue>DBPTK</objectIdentifierValue>
  </objectIdentifier>
  <environmentFunction>
    <environmentFunctionType>software</environmentFunctionType>
    <environmentFunctionLevel>1</environmentFunctionLevel>
  </environmentFunction>
  <environmentFunction>
    <environmentFunctionType>
      software application
    </environmentFunctionType>
    <environmentFunctionLevel>2</environmentFunctionLevel>
  </environmentFunction>
  <environmentDesignation>
    <environmentName>DBPTK Desktop</environmentName>
    <environmentVersion>2.5.3</environmentVersion>
    <environmentDesignationNote>
      Desktop application to store database to archival format, validate it
      and browse the content.
      https://github.com/keeps/dbptk-ui/releases/tag/v2.5.3
    </environmentDesignationNote>
  </environmentDesignation>
<!-- PREMIS file continues but further elements left out in this listing-->
</object>
```

#### 4.5.2.2 *Establishing the relationship between a representation and creation software*

In order to establish a connection between the representation format which has been created and the creating software, it is necessary to use the semantic component named “relationship” which groups a number of semantic units describing different information constituting the relationship. The relationship element can bind both non-environmental objects together with environmental objects, and it can bind environmental objects together with other environmental objects. The following example shows how the IP representation format MUST be related to the Access Software:

```
<object xsi:type="representation">
```

---

```

    <objectIdentifier>
      <objectIdentifierType>filepath</objectIdentifierType>
      <objectIdentifierValue>
        representations\AVID.SA.18006.rep0
      </objectIdentifierValue>
    </objectIdentifier>
    <significantProperties>
      <significantPropertiesType>
        IP representation format
      </significantPropertiesType>
      <significantPropertiesValue>SIARD2</significantPropertiesValue>
    </significantProperties>
<!-- The following is the relation between the software and the IP representation -->
    <relationship>
      <relationshipType>dependency</relationshipType>
      <relationshipSubType>is Source Of</relationshipSubType>
      <relatedObjectIdentifier>
        <relatedObjectIdentifierType>
          Local
        </relatedObjectIdentifierType>
        <relatedObjectIdentifierValue>
          DBPTK
        </relatedObjectIdentifierValue>
      </relatedObjectIdentifier>
      <relatedEnvironmentPurpose>software</relatedEnvironmentPurpose>
    </relationship>
<!-- PREMIS file continues but further elements left out in this listing-->
</object>

```

The nature of the relationship, `<relationshipType>` is used (value, e.g. “dependency”). Intimately linked to this is also the indication of a `<relationshipSubType>` (e.g. “is Source Of”).

To identify the software, which is used to create the representation, the `<relatedObjectIdentifier>` is employed; and the `<relatedEnvironmentPurpose>` provides a hint about what the purpose is (here: to “software”).

Since it is not always possible to create the IP representation formats with one piece of software, it can be necessary to model software dependencies and sequences between several pieces of software in PREMIS.

Observe that the creation events can be stored using the event entity.

#### 4.5.3 Metadata regarding Representations and Access Software in a DIP

When a digital object is requested from the repository by a researcher or another repository, the knowledge of how to render the object is critical. The delivery to the researcher can be made in many ways; we only describe the metadata that should be included.



One example of a common research request is to get access to a database. When delivering the database to the researcher, it is important to give information on how the researcher should render the database. This information is in this example given to the researcher with the help of two different entities in PREMIS. One entity is an object with the type representation, which describes the “set of files, including structural metadata” composing the database. The other object entity is in the form of an intellectual entity of the type environment describing the software to use for rendering the database.

One of the core concepts in PREMIS is the definition of a representation, but it is also important to note that when using the CSIP structure, there is also an incorporation of physical management of different representations. When implementing PREMIS in CSIPs and especially in a DIP, it is necessary to decide if PREMIS files exist at representation level or at root level only (see more in the CSIP), and how fine-grained each description should be.

Three pieces of information are needed to enable rendition of the digital objects in the DIP, which are expressed with the help of PREMIS:

- The first identifies the representation to be rendered;
- The second identifies the software to enable the rendering;
- The third establishes the relationship between what is rendered and the software performing the rendering.

#### 4.5.3.1 *Identifying the representation to be rendered*

To describe the specific representation format in PREMIS, the available semantic component (a grouping unit) for objects named “significantProperties” MUST be used. An example is shown below.

```
<object xsi:type="representation">
  <objectIdentifier>
    <objectIdentifierType>filepath</objectIdentifierType>
  <objectIdentifierValue>
    representations\AVID.SA.18006.rep0\
  </objectIdentifierValue>
</objectIdentifier>
<significantProperties>
  <significantPropertiesType>
    Representation format
  </significantPropertiesType>
  <significantPropertiesValue>SIARD2</significantPropertiesValue>
</significantProperties>
<!-- PREMIS file continues but further elements left out in this listing-->
</object>
```

Note that the object type is “representation” and that the objectIdentifierType value is using the example type “filepath”, which is local to the IP scope value. The objectIdentifierValue is the file path to the representation folder or could be a file path to a file.

#### 4.5.3.2 *Identifying the software for rendering*

With the release of PREMIS 3.0 a description of an environment is seen as an intellectual entity type object, which means that both non-environmental objects and environmental objects exist and can be described with PREMIS. The access software is, therefore, an environmental object which per default is an intellectual entity following the definitions in PREMIS. In PREMIS a semantic unit named “environmentFunction” is available for objects which describes the environment object(s) functions with different levels of granularity. It is suggested to use the vocabulary available and recommended in PREMIS for describing the environment function <https://id.loc.gov/vocabulary/preservation/environmentFunctionType.html>. When describing an environment, it is recommended that the semantic unit “environmentDesignation” is used for information identifying the environment by using human-readable language, which can be expected to be understood outside of a digital repository. See the example below which **MUST** be followed to describe the software following this vocabulary:

```
<object xsi:type="intellectualEntity">
  <objectIdentifier>
    <objectIdentifierType>local</objectIdentifierType>
    <objectIdentifierValue>DBPTK</objectIdentifierValue>
  </objectIdentifier>
  <environmentFunction>
    <environmentFunctionType>software</environmentFunctionType>
    <environmentFunctionLevel>1</environmentFunctionLevel>
  </environmentFunction>
  <environmentFunction>
    <environmentFunctionType>
      software application
    </environmentFunctionType>
    <environmentFunctionLevel>2</environmentFunctionLevel>
  </environmentFunction>
  <environmentDesignation>
    <environmentName>DBPTK Desktop</environmentName>
    <environmentVersion>2.5.3</environmentVersion>
    <environmentDesignationNote>
      Desktop application to store database to archival format, validate it
      and browse the content.
      https://github.com/keeps/dbptk-ui/releases/tag/v2.5.3
    </environmentDesignationNote>
  </environmentDesignation>
<!-- PREMIS file continues but further elements left out in this listing-->
</object>
```

#### 4.5.3.3 *Establishing the relationship between a representation and rendering software*

In order to establish a connection between the representation format to be rendered and the access software to render it, it is necessary to use the semantic component named “relationship” which groups a number of semantic units describing different information constituting the relationship. The relationship element can bind both non-environmental

objects together with environmental objects, and it can bind environmental objects together with other environmental objects. The following example shows how the IP representation format MUST be related to the Access Software:

```
<object xsi:type="representation">
  <objectIdentifier>
    <objectIdentifierType>filepath</objectIdentifierType>
    <objectIdentifierValue>
      representations\AVID.SA.18006.rep0
    </objectIdentifierValue>
  </objectIdentifier>
  <significantProperties>
    <significantPropertiesType>
      IP representation format
    </significantPropertiesType>
    <significantPropertiesValue>SIARD2</significantPropertiesValue>
  </significantProperties>
  <!-- The following is the relation between the software and the IP representation -->
  <relationship>
    <relationshipType>dependency</relationshipType>
    <relationshipSubType>requires</relationshipSubType>
    <relatedObjectIdentifier>
      <relatedObjectIdentifierType>
        Local
      </relatedObjectIdentifierType>
      <relatedObjectIdentifierValue>
        DBPTK
      </relatedObjectIdentifierValue>
    </relatedObjectIdentifier>
    <relatedEnvironmentPurpose>render</relatedEnvironmentPurpose>
  </relationship>
  <!-- PREMIS file continues but further elements left out in this listing-->
</object>
```

The nature of the relationship, `<relationshipType>` is used (value, e.g. “dependency”). Intimately linked to this is also the indication of a `<relationshipSubType>` (e.g. “requires”). In order to identify the access software, which is used to render the representation, the `<relatedObjectIdentifier>` is employed; and the `<relatedEnvironmentPurpose>` provides a hint about what the purpose is (here: to “render”). Since it is not always possible to render the IP representation formats with one piece of access software, it can be necessary to model software dependencies and sequences between several pieces of software in PREMIS.

## 5 Metadata in the Information Package

The preservation metadata is placed in the administrative section of the information package.

## 5.1 Using the Common Specification for Preservation Metadata together with the Common Specification for Information Packages (CSIP)

When the preservation metadata documents have been created, they can be transmitted in and stored in a package following the requirements described in the chapter about Use of the METS administrative metadata section (element amdSec) in the CSIP specification <http://earkcsip.dilcis.eu/>.

There is one “Digital Provenance metadata” section present for each file of the type preservation metadata in the transfer or storage. The requirements pertaining to the file itself like mimetype, size and equals are not described in the following table.

### 5.1.1 Specific fields to use in CSIP for the preservation metadata

The preservation metadata is encoded with the help of PREMIS as previously described.

**Table 1: Specific fields to use in CSIP for a PREMIS document**

Element name	METS path	Value
Administrative metadata	mets/amdSec	n/a One amdSec is present
Digital provenance metadata	mets/amdSec/digiprovMD	One digiprovMD for each piece of PREMIS metadata
Type of metadata	mets/amdSec/digiprovMD/mdRef/@MDTYPE	PREMIS

## 5.2 Placement of preservation metadata documents in a CSIP Information Package

All the documents giving the preservation metadata for the data objects in the transfer are placed in the “metadata” section of the IP (see Figure 1 and Figure 2).

## 6 Metadata

In the following tables, the different parts of the PREMIS document will be described.

### 6.1 Use of PREMIS

The main requirement for PREMIS files in a CSIP Information Package is that these need to follow the official PREMIS Schema version 3.0 <http://www.loc.gov/standards/premis/premis.xsd> (by CS Preservation Metadata used version in May 2020). As new versions of the PREMIS Schema become available, the DILCIS Board will evaluate these and, if necessary, update the CS Preservation Metadata, respectively.

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

The PREMIS XML schema only allows metadata to be embedded. This contrasts with the CSIP, which allows both the embedding of metadata but, for scalability concerns, recommends the use of referencing. For the preservation metadata, the inclusion follows the PREMIS specification.

The rest of this section describes the use of the following PREMIS elements: `premis`, `object`, `event`, `agent` and `rights`. For each element, the restrictions imposed by the CS preservation Metadata are explained in addition to those in the official PREMIS documentation. An implementation of this specification can extend the restrictions in line with requirements specific to the implementation. In these cases, follow the PREMIS documentation and create an implementation that uses the CS Preservation Metadata as the starting point.

The specification also uses the method of putting all PREMIS elements into one XML document combining objects, events, agents, and rights that belongs together to one unit. In the implementation in a system, it is possible to have these sections being in their own files and the joint document being created when the PREMIS document being part of the transmission is created.

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

#### 6.1.1 Use of the PREMIS root element (element `premis`)

The PREMIS XML document's root element (`<premis>`) describes the container for the information being preserved. The `<premis>` element of a CS Preservation Metadata conformant PREMIS XML-document uses attributes from the PREMIS XML-schema specification.

As well as the PREMIS attributes, the PREMIS document's root `<premis>` element must define all of the relevant namespaces and locations for the 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" ([https://www.w3schools.com/xml/schema\\_schema.asp](https://www.w3schools.com/xml/schema_schema.asp)). The schema identifiers and locations for a typical CS Preservation Metadata `<premis>` element is shown below.

```
<premis xmlns=http://www.loc.gov/premis/v3
        xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
        xsi:schemaLocation="http://www.loc.gov/premis/v3
http://www.loc.gov/standards/premis/premis.xsd
        version="3.0">
```

##### 6.1.1.1 Location of XML schema

When using XML schemas, the availability of the actual schema resources should be considered. Externally hosted resources are not guaranteed to be available in the future or in restricted operating environments. It is recommended that copies of all XML schema resources should be included in the information package, located in appropriate "schemas" folders at the 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.

Table 2: PREMIS root element

ID	Name and location	Description and usage	Cardinality Level
PM1	PREMIS version premis/@version="3.0"	The version of PREMIS is given in the root element.	1..1 MUST

### 6.1.2 Metadata for an object of entity type "Intellectual Entity" and specifically of type "Environment"

For each software needed to render the digital objects described, one object is created, which is called an intellectual entity of the specific type environment.

Table 3: Description of the software for rendering

ID	Name and location	Description and usage	Cardinality Level
PM2	Object category object[@xsi:type='intellectualEntity']	The object category is mandatory and follows the vocabulary "Object Category" defined in PREMIS.  See also: <a href="https://id.loc.gov/vocabulary/preservation/objectCategory.html">https://id.loc.gov/vocabulary/preservation/objectCategory.html</a>	1..1 MUST
PM3	Object identification object/objectIdentifier	The object is given an identification to be easily identifiable.	1..1 MUST
PM4	Type of identification object/objectIdentifier/objectIdentifierType	The identification follows a specific schema. If the identification is locally created the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
PM5	Identification value object/objectIdentifier/objectIdentifierValue	The identification value expressed according to the identification type given in the "Type of identification" element.	1..1 MUST
PM6	Environment function description object/environmentFunction	A grouping element for the description of the environment function.	1..n MUST

<b>PM7</b>	Environment function type  object/environmentFunction/environmentFunctionType	The function of the environment is given with a value from the vocabulary "Environment Function Type".  See also: <a href="https://id.loc.gov/vocabulary/preservation/environmentFunctionType.html">https://id.loc.gov/vocabulary/preservation/environmentFunctionType.html</a>	1..1  MUST
<b>PM8</b>	Environment level  object/environmentFunction/environmentFunctionLevel	An integer giving the sequence number for when the environment is used in the rendering of the digital object.	1..1  MUST
<b>PM9</b>	Environment description  object/environmentDesignation	A description of the environment.	1..1  MUST
<b>PM10</b>	Environment name  object/environmentDesignation/environmentName	The name of the software which is described. PREMIS recommends using an established value list of names at least in the organisation.	1..1  MUST
<b>PM11</b>	Environment version  object/environmentDesignation/environmentVersion	If possible, give the version of the software.	0..1  SHOULD
<b>PM12</b>	Environment Origin  object/environmentDesignation/environmentOrigin	If possible, give the origination of the software (i.e. "Microsoft"). PREMIS recommends using an established value list of names at least in the organisation.	0..1  SHOULD
<b>PM13</b>	Environment note  object/environmentDesignation/environmentDesignationNote	If extra information about the software is available, it can be expressed in plain text in a note.	0..1  MAY

Example of a description of a software.

```
<object xsi:type="intellectualEntity">
  <objectIdentifier>
    <objectIdentifierType>local</objectIdentifierType>
    <objectIdentifierValue>DBPTK</objectIdentifierValue>
  </objectIdentifier>
  <environmentFunction>
    <environmentFunctionType>software</environmentFunctionType>
    <environmentFunctionLevel>1</environmentFunctionLevel>
  </environmentFunction>
```

```

<environmentFunction>
  <environmentFunctionType>
    software application
  </environmentFunctionType>
  <environmentFunctionLevel>2</environmentFunctionLevel>
</environmentFunction>
<environmentDesignation>
  <environmentName>DBPTK Desktop</environmentName>
  <environmentVersion>2.5.3</environmentVersion>
  <environmentDesignationNote>
    Desktop application to store database to archival format, validate it and browse the
    content. https://github.com/keeps/dbptk-ui/releases/tag/v2.5.3
  </environmentDesignationNote>
</environmentDesignation>
</object>

```

### 6.1.3 Metadata for an object of entity type “Representation”

For each digital object created or needing rendering, one object is created with the entity type representation.

Table 4: Description of the data object for creation or rendering

ID	Name and location	Description and usage	Cardinality Level
PM14	Object category object[@xsi:type='representation']	The object category is mandatory and follows the vocabulary “Object Category” defined in PREMIS.  See also: <a href="https://id.loc.gov/vocabulary/preservation/objectCategory.html">https://id.loc.gov/vocabulary/preservation/objectCategory.html</a>	1..1 MUST
PM15	Object identification object/objectIdentifier	The object is given an identification to be easily identifiable.	1..1 MUST
PM16	Type of identification object/objectIdentifier/objectIdentifierType	The identification is following a specific schema. If the identification is created locally, the value “local” is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
PM17	Identification value object/objectIdentifier/objectIdentifierValue	The identification value expressed according to the identification type given in the “Type of identification” element.	1..1 MUST
PM18	Significant properties	The object might carry significant properties that can be expressed.	0..n SHOULD



	object/significantProperties		
<b>PM19</b>	Type of significant property object/significantProperties/significantPropertiesType	A definition of the significant property being expressed. Examples include content, structure, behaviour, page count, page width, typeface, hyperlinks and image count.	1..1 MUST
<b>PM20</b>	Value of the significant property object/significantProperties/significantPropertiesValue	The value of the significant property.	1..1 MUST
<b>PM21</b>	Relationship with the software object/relationship	The connection between a digital object and the rendering software is established with relationships.	1..n MUST
<b>PM22</b>	Type of relationship object/relationship/relationshipType	The type of relationship is defined with the vocabulary "Relationship Type"  See also: <a href="https://id.loc.gov/vocabulary/preservation/relationshipType.html">https://id.loc.gov/vocabulary/preservation/relationshipType.html</a>	1..1 MUST
<b>PM23</b>	The subtype of the relationship object/relationship/relationshipSubType	The specific type of relationship is defined with the vocabulary "Relationship Subtype".  See also: <a href="https://id.loc.gov/vocabulary/preservation/relationshipSubType.html">https://id.loc.gov/vocabulary/preservation/relationshipSubType.html</a>	1..1 MUST
<b>PM24</b>	Identifier to related object object/relationship/relatedObjectIdentifier	The object describing the software used for rendering the data is identified with the identifier it has been given.	1..1 MUST
<b>PM25</b>	Type of identifier object/relationship/relatedObjectIdentifier/relatedObjectIdentifierType	The type of the related object identifier following a vocabulary. The value "local" is used if the identification has been created locally.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
<b>PM26</b>	Identification of the related object object/relationship/relatedObjectIdentifier/relatedObjectIdentifierValue	The identification value of the object describing the rendering software.	1..1 MUST

<b>PM27</b>	Purpose of the related environment object/relationship/relatedEnvironmentPurpose	A term describing the purpose of the software used for rendering the digital object. The term is defined in the vocabulary “Event Type”.  See also: <a href="https://id.loc.gov/vocabulary/preservation/eventType.html">https://id.loc.gov/vocabulary/preservation/eventType.html</a>	0..1  SHOULD
-------------	---	--	--------------------

Example of creating the connection between a digital object and the rendering software.

```
<object xsi:type="representation">
  <objectIdentifier>
    <objectIdentifierType>filepath</objectIdentifierType>
    <objectIdentifierValue>
      representations\AVID.SA.18006.rep0
    </objectIdentifierValue>
  </objectIdentifier>
  <significantProperties>
    <significantPropertiesType>
      IP representation format
    </significantPropertiesType>
    <significantPropertiesValue>SIARD2</significantPropertiesValue>
  </significantProperties>
  <!-- The following is the relation between the software and the IP representation -->
  <relationship>
    <relationshipType>dependency</relationshipType>
    <relationshipSubType>requires</relationshipSubType>
    <relatedObjectIdentifier>
      <relatedObjectIdentifierType>
        Local
      </relatedObjectIdentifierType>
      <relatedObjectIdentifierValue>
        DBPTK
      </relatedObjectIdentifierValue>
    </relatedObjectIdentifier>
    <relatedEnvironmentPurpose>render</relatedEnvironmentPurpose>
  </relationship>
</object>
```

#### 6.1.4 Metadata for object of entity type “File”

For each digital object being preserved, one object is created with the entity type file. It is also possible to do the description at an aggregated level.

Table 5: Description of the data object of entity type file

ID	Name and location	Description and usage	Cardinality Level
PM28	Object category object[@xsi:type=file]	The object category is mandatory and follows the vocabulary "Object Category" defined in PREMIS.  See also: <a href="https://id.loc.gov/vocabulary/preservation/objectCategory.html">https://id.loc.gov/vocabulary/preservation/objectCategory.html</a>	1..1  MUST
PM29	Object identification object/objectIdentifier	The object is given an identification to facilitate easy identification.	1..n  MUST
PM30	Type of identification object/objectIdentifier/objectIdentifierType	The identification follows a specific schema. If the identification is locally created, the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1  MUST
PM31	Identification value object/objectIdentifier/objectIdentifierValue	The identification value expressed according to the identification type given in the "Type of identification" element.	1..1  MUST
PM32	Object characteristics object/objectCharacteristics	The object has characteristics that are important for the preservation of the digital object.	1..n  MUST
PM33	Fixity of the digital object object/objectCharacteristics/fixity	The checksum of the digital object can be stored.	0..n  SHOULD
PM34	Fixity algorithm object/objectCharacteristics/fixity/messageDigestAlgorithm	The algorithm used for calculating the checksum according to vocabulary named "Cryptographic Hash Functions".  See also: <a href="https://id.loc.gov/vocabulary/preservation/cryptographicHashFunctions.html">https://id.loc.gov/vocabulary/preservation/cryptographicHashFunctions.html</a>	1..1  MUST
PM35	Fixity value object/objectCharacteristics/fixity/messageDigest	The calculated checksum.	1..1  MUST
PM36	Fixity originator object/objectCharacteristics/fixity/messageDigestOriginator	The name of the software used to calculate the checksum can be given.	0..1  MAY

<b>PM37</b>	File format object/objectCharacteristics/format	The file format of the file should be given by either using the file format name grouping or the file format registry grouping.	0..1 SHOULD
<b>PM38</b>	File format name grouping object/objectCharacteristics/format/formatDesignation	Grouping element for file format name and version.	0..1 SHOULD
<b>PM39</b>	File format name object/objectCharacteristics/format/formatDesignation/formatName	File format name.	1..1 MUST
<b>PM40</b>	File format version object/objectCharacteristics/format/formatDesignation/formatVersion	File format version.	0..1 SHOULD
<b>PM41</b>	File format registry grouping object/objectCharacteristics/format/formatRegistry	Grouping element for linking to a file format registry.	0..1 SHOULD
<b>PM42</b>	File format registry name object/objectCharacteristics/format/formatRegistry/formatRegistryName	The name of the file format registry used. The recommendation is to use the PRONOM registry.  See also: <a href="http://www.nationalarchives.gov.uk/PRONOM/Default.aspx">http://www.nationalarchives.gov.uk/PRONOM/Default.aspx</a>	1..1 MUST
<b>PM43</b>	File format registry key object/objectCharacteristics/format/formatRegistry/formatRegistryKey	The key for the file format in the registry given in the file format registry name.	1..1 MUST
<b>PM44</b>	File format registry role object/objectCharacteristics/format/formatRegistry/formatRegistryRole	The role of the file format registry can be expressed using the vocabulary "Format Registry Role".  See also: <a href="https://id.loc.gov/vocabulary/preservation/formatRegistryRole.html">https://id.loc.gov/vocabulary/preservation/formatRegistryRole.html</a>	0..1 MAY
<b>PM45</b>	Creating application of a file object/objectCharacteristics/creatingApplication/	It is possible to add information about the application that has created a file.	0..n MAY

<b>PM46</b>	Name of creating application  object/objectCharacteristics/creatingApplication/creatingApplicationName	The name of the application that has created the file object.	1..1  MUST
<b>PM47</b>	Version of creating application  object/objectCharacteristics/creatingApplication/creatingApplicationVersion	The version of the application that has created the file object.	0..1  MAY
<b>PM48</b>	Date of creation by the application  object/objectCharacteristics/creatingApplication/dateCreatedByApplication	The date the application created the file object.	0..1  MAY
<b>PM49</b>	Extended information regarding creating application  object/objectCharacteristics/creatingApplication/creatingApplicationExtension	It is possible to provide information in another XML format regarding the creating application.	0..n  MAY
<b>PM50</b>	Extended information about the characteristics  object/objectCharacteristics/objectCharacteristicsExtension	If the file format has been evaluated using JHove the outcome of the evaluation can be placed in the extension element.	0..1  MAY
<b>PM51</b>	Original name  object/originalName	The original name can be given if the name has been changed during the preservation actions.	0..1  SHOULD
<b>PM52</b>	Storage  object/storage	The storage location of the digital object can be described.	0..n  MAY
<b>PM53</b>	The location of the content  object/storage/contentLocation	The location of the digital content can be given for providing easy access.	0..1  COULD
<b>PM54</b>	Type of location  object/storage/contentLocation/contentLocationType	The type of location is given with values from the vocabulary "Content Location Type".  See also: <a href="https://id.loc.gov/vocabulary/preservation/contentLocationType.html">https://id.loc.gov/vocabulary/preservation/contentLocationType.html</a>	1..1  MUST

<b>PM55</b>	Location object/storage/contentLocation/contentLocationValue	The name/identifier of the location of the content.	1..1 MUST
<b>PM56</b>	Storage medium object/storage/storageMedium	The medium of the storage can be given using the vocabulary "Storage Medium".  See also: <a href="https://id.loc.gov/vocabulary/preservation/storageMedium.html">https://id.loc.gov/vocabulary/preservation/storageMedium.html</a>	0..1 MAY
<b>PM57</b>	Relationship with other objects or events object/relationship	The connection between a digital object and other digital objects and/or events is established with relationships.	0..n SHOULD
<b>PM58</b>	Type of relationship object/relationship/relationshipType	The type of relationship is defined with the vocabulary "Relationship Type".  See also: <a href="https://id.loc.gov/vocabulary/preservation/relationshipType.html">https://id.loc.gov/vocabulary/preservation/relationshipType.html</a>	1..1 MUST
<b>PM59</b>	The subtype of the relationship object/relationship/relationshipSubType	The specific type of relationship is defined with the vocabulary "Relationship Subtype".  See also: <a href="https://id.loc.gov/vocabulary/preservation/relationshipSubType.html">https://id.loc.gov/vocabulary/preservation/relationshipSubType.html</a>	1..1 MUST
<b>PM60</b>	Identifier to related object object/relationship/relatedObjectIdentifier	The object describing the related digital object with the identifier it has been given.	1..n MUST
<b>PM61</b>	Type of identifier object/relationship/relatedObjectIdentifier/relatedObjectIdentifierType	The type of the related object identifier following a vocabulary. If the identification has been locally created, the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
<b>PM62</b>	Identification of the related object object/relationship/relatedObjectIdentifier/relatedObjectIdentifierValue	The identification value of the object describing the related digital object.	1..1 MUST
<b>PM63</b>	Identifier to related event object/relationship/relatedEventIdentifier	The event related to the digital object with the identifier it has been given.	0..n SHOULD

<b>PM64</b>	Type of identifier object/relationship/relatedEventIdentifier/relatedObjectIdentifierType	The type of the related event identifier following a vocabulary. If the identification has been locally created, the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1  MUST
<b>PM65</b>	Identification of the related event object/relationship/relatedEventIdentifier/relatedEventIdentifierValue	The identification value of the event describing the related event.	1..1  MUST
<b>PM66</b>	Rights statement identification object/linkingRightsStatementIdentifier	The digital object can relate to one or more rights statements.	0..n  COULD
<b>PM67</b>	Rights statement identifier type object/linkingRightsStatementIdentifier/linkingRightsStatementIdentifierType	The type of rights statement identifier following a vocabulary. If the identification has been locally created, the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1  MUST
<b>PM68</b>	Rights statement identifier value object/linkingRightsStatementIdentifier/linkingRightsStatementIdentifierValue	The identification value of the right statements connected with the digital object.	1..1  MUST

Example of a file object.

```
<object xsi:type="file">
  <objectIdentifier>
    <objectIdentifierType>local</objectIdentifierType>
    <objectIdentifierValue>fileId001</objectIdentifierValue>
  </objectIdentifier>
  <objectIdentifier>
    <objectIdentifierType>filepath</objectIdentifierType>
    <objectIdentifierValue>metadata/file.xml</objectIdentifierValue>
  </objectIdentifier>
  <objectCharacteristics>
    <fixity>
      <messageDigestAlgorithm>SHA-256</messageDigestAlgorithm>
      <messageDigest>3b1d00f7871d9102001c77f...</messageDigest>
      <messageDigestOriginator>/usr/bin/sha256sum</messageDigestOriginator>
    </fixity>
  </objectCharacteristics>
</object>
```

---

```
<format>
  <formatDesignation>
    <formatName>XML</formatName>
    <formatVersion>1.0</formatVersion>
  </formatDesignation>
  <formatRegistry>
    <formatRegistryName>PRONOM</formatRegistryName>
    <formatRegistryKey>fmt/101</formatRegistryKey>
    <formatRegistryRole>specification</formatRegistryRole>
  </formatRegistry>
</format>
<objectCharacteristicsExtension>
  <jhove>
    ...
  </jhove>
</objectCharacteristicsExtension>
</objectCharacteristics>
<originalName>originalfilename.ext</originalName>
<storage>
  <contentLocation>
    <contentLocationType>Uniform Resource Identifier</contentLocationType>
    <contentLocationValue>
      /path/to/file.txt
    </contentLocationValue>
  </contentLocation>
  <storageMedium>hard disk HD2253</storageMedium>
</storage>
<relationship>
  <relationshipType>structural</relationshipType>
  <relationshipSubType>is included in</relationshipSubType>
  <relatedObjectIdentifier>
    <relatedObjectIdentifierType>local</relatedObjectIdentifierType>
    <relatedObjectIdentifierValue>
      ID123e4567-e89b-12d3-a456-426655440000
    </relatedObjectIdentifierValue>
  </relatedObjectIdentifier>
  <relatedEventIdentifier>
    <relatedEventIdentifierType>local</relatedEventIdentifierType>
    <relatedEventIdentifierValue>
      ingest-001
    </relatedEventIdentifierValue>
  </relatedEventIdentifier>
</relationship>
<linkingRightsStatementIdentifier>
  <linkingRightsStatementIdentifierType>
    filepath
  </linkingRightsStatementIdentifierType>
```



```

<linkingRightsStatementIdentifierValue>
  metadata/file.xml
</linkingRightsStatementIdentifierValue>
</linkingRightsStatementIdentifier>
</object>

```

### 6.1.5 Metadata for an agent

When there are events stored pertaining to the digital objects, the agents performing the events are described.

Table 6: Description of the agent

ID	Name and location	Description and usage	Cardinality Level
PM69	Agent agent	An agent connected with an event carried out involving a digital object.	0..n SHOULD
PM70	Agent identification agent/agentIdentifier	The agent is given an identification to be easily identifiable.	1..n MUST
PM71	Type of identification agent/agentIdentifier/agentIdentifierType	The identification follows a specific schema. If the identification is locally created, the value “local” is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
PM72	Identification value agent/agentIdentifier/agentIdentifierValue	The identification value expressed according to the identification type given in the “Type of identification” element.	1..1 MUST
PM73	Agent name agent/agentName	The name of the agent in a human-understandable form.	1..1 MUST
PM74	Agent type agent/agentType	The specific type of agent is defined with the vocabulary “Agent Type”.  See also: <a href="https://id.loc.gov/vocabulary/preservation/agentType.html">https://id.loc.gov/vocabulary/preservation/agentType.html</a>	1..1 MUST
PM75	Agent version agent/agentVersion	If the agent is software, then the version of the software can be given.	0..1 SHOULD
PM76	Agent Note agent/agentNote	The agent can be described with a short text.	0..1 MAY

<b>PM77</b>	Rights for the agent agent/linkingRightStatementIdentifier	When rights are granted for the agent to carry out an event, the agent is linked to those rights statements.	0..n SHOULD
<b>PM78</b>	Rights statement identifier type agent/linkingRightsStatementIdentifier/linkingRightsStatementIdentifierType	The type of rights statement identifier following a vocabulary. If the identification has been locally created the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
<b>PM79</b>	Rights statement identifier value agent/linkingRightsStatementIdentifier/linkingRightsStatementIdentifierValue	The identification value of the right statements connected with the digital object.	1..1 MUST

Example of a description of an agent.

```
<agent>
  <agentIdentifier>
    <agentIdentifierType>local</agentIdentifierType>
    <agentIdentifierValue>Indexer</agentIdentifierValue>
  </agentIdentifier>
  <agentName>Indexing Software</agentName>
  <agentType>Software</agentType>
  <linkingRightsStatementIdentifier>
    <linkingRightsStatementIdentifierType>
      local
    </linkingRightsStatementIdentifierType>
    <linkingRightsStatementIdentifierValue>
      discovery-right-001
    </linkingRightsStatementIdentifierValue>
  </linkingRightsStatementIdentifier>
</agent>
```

**6.1.6 Metadata for an event**

When there are events stored pertaining to the digital objects, all the events are stored as separate events in each own event.

Table 7: Description of the event

ID	Name and location	Description and usage	Cardinality Level
<b>PM80</b>	Event	Each event carried out involving a digital object is recorded.	0..n

	event		SHOULD
<b>PM81</b>	Event identification event/eventIdentifier	The event is given an identification to be easily identifiable.	1..n MUST
<b>PM82</b>	Type of identification event/eventIdentifier/eventIdentifierType	The identification is following a specific schema. If the identification is locally created, the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
<b>PM83</b>	Identification value event/eventIdentifier/eventIdentifierValue	The identification value expressed according to the identification type given in the "Type of identification" element.	1..1 MUST
<b>PM84</b>	Event type event/eventType	The specific type of event is defined with the vocabulary "Event Type".  See also: <a href="https://id.loc.gov/vocabulary/preservation/eventType.html">https://id.loc.gov/vocabulary/preservation/eventType.html</a>	1..1 MUST
<b>PM85</b>	Time for the event event/eventDateTime	The time and date the event was carried out.  See also: <a href="https://id.loc.gov/datatypes/edtf.html">https://id.loc.gov/datatypes/edtf.html</a>	1..1 MUST
<b>PM86</b>	Outcome of the event event/eventOutcomeInformation/eventOutcome	The outcome of the event is defined with the vocabulary "Event Outcome".  See also: <a href="https://id.loc.gov/vocabulary/preservation/eventOutcome.html">https://id.loc.gov/vocabulary/preservation/eventOutcome.html</a>	1..1 MUST
<b>PM87</b>	Agent performing the event event/linkingAgentIdentifier	The agent carrying out the event.	0..n SHOULD
<b>PM88</b>	Agent identifier type event/linkingAgentIdentifier/linkingAgentIdentifierType	The type of agent identifier following a vocabulary. The value "local" is used if the identification has been locally created.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
<b>PM89</b>	Agent identifier value event/linkingAgentIdentifier/linkingAgentIdentifierValue	The identification value of the agent connected with the event.	1..1 MUST
<b>PM90</b>	Object affected by the event	The object the event has affected.	0..n

	event/linkingObjectIdentifier		SHOULD
<b>PM91</b>	Object identifier type event/linkingObjectIdentifier/linkingObjectIdentifierType	The type of the object identifier following a vocabulary. If the identification has been locally created the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1  MUST
<b>PM92</b>	Object identifier value event/linkingObjectIdentifier/linkingObjectIdentifierValue	The identification value of the object connected with the event.	1..1  MUST

Example of a description of an event.

```

<event>
  <eventIdentifier>
    <eventIdentifierType>local</eventIdentifierType>
    <eventIdentifierValue>migration-001</eventIdentifierValue>
  </eventIdentifier>
  <eventType>MIGRATION</eventType>
  <eventDateTime>2015-09-01T01:00:00+01:00</eventDateTime>
  <eventOutcomeInformation>
    <eventOutcome>success</eventOutcome>
  </eventOutcomeInformation>
  <linkingAgentIdentifier>
    <linkingAgentIdentifierType>local</linkingAgentIdentifierType>
    <linkingAgentIdentifierValue>
      FileFormatConversion001
    </linkingAgentIdentifierValue>
  </linkingAgentIdentifier>
  <linkingObjectIdentifier>
    <linkingObjectIdentifierType>filepath</linkingObjectIdentifierType>
    <linkingObjectIdentifierValue>
      metadata/file.xml
    </linkingObjectIdentifierValue>
  </linkingObjectIdentifier>
</event>

```

### 6.1.7 Metadata for rights

When there are rights stored pertaining to the digital objects, the rights are described.

Table 8: Description of the rights

ID	Name and location	Description and usage	Cardinality Level
PM93	Rights rights	All rights statements pertaining to the digital object and the agents are described using the rights element.	0..1 SHOULD
PM94	Rights statements rights/rightsStatement	Each rights statement is placed in a rights statement element.	1..n MUST
PM95	Rights identification rights/rightsStatement/rightsStatementIdentifier	The right statement is given an identification to be easily identifiable.	1..n MUST
PM96	Type of identification rights/rightsStatement/rightsStatementIdentifier/rightsStatementIdentifierType	The identification is following a specific schema. If the identification is created locally the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
PM97	Identification value rights/rightsStatement/rightsStatementIdentifier/rightsStatementIdentifierValue	The identification value expressed according to the identification type given in the "Type of identification" element.	1..1 MUST
PM98	Rights type rights/rightsStatement/rightsBasis	The specific type of rights is defined with the vocabulary "Rights Basis".  See also: <a href="https://id.loc.gov/vocabulary/preservation/rightsBasis.html">https://id.loc.gov/vocabulary/preservation/rightsBasis.html</a>	1..1 MUST
PM99	Copyright information rights/rightsStatement/[rightsBasis='copyright'] rights/rightsStatement/copyrightInformation	When rights basis is copyright these elements occur.	0..1 SHOULD
PM100	Copyright status rights/rightsStatement/copyrightInformation/copyrightStatus	The specific type of the copyrights is defined with the vocabulary "Copyright Status".  See also: <a href="https://id.loc.gov/vocabulary/preservation/copyrightStatus.html">https://id.loc.gov/vocabulary/preservation/copyrightStatus.html</a>	1..1 MUST
PM101	Copyright jurisdiction	The country in which the copyright law is applied. Use a value from ISO 3166.	1..1 MUST

	rights/rightsStatement/copyrightInformation/copyrightJurisdiction	See also: <a href="https://www.iso.org/iso-3166-country-codes.html">https://www.iso.org/iso-3166-country-codes.html</a>	
<b>PM102</b>	Copyright documentation rights/rightsStatement/copyrightInformation/copyrightDocumentationIdentifier	Copyright documentation can be linked to give more information.	0..1 MAY
<b>PM103</b>	Copyright documentation identifier type rights/rightsStatement/copyrightInformation/copyrightDocumentationIdentifier/copyrightDocumentationIdentifierType	The type of the identifier following a vocabulary. If the identification has been locally created the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
<b>PM104</b>	Copyright documentation identifier value rights/rightsStatement/copyrightInformation/copyrightDocumentationIdentifier/copyrightDocumentationIdentifierValue	The identification value of the copyright documentation.	1..1 MUST
<b>PM105</b>	License information rights/rightsStatement/[rightsBasis='license'] rights/rightsStatement/licenseInformation	When rights basis is license these elements occur.	0..1 SHOULD
<b>PM106</b>	License documentation rights/rightsStatement/licenseInformation/licenseDocumentationIdentifier	It is possible to link to license documentation to provide more information.	0..1 MAY
<b>PM107</b>	License documentation identifier type rights/rightsStatement/licenseInformation/licenseDocumentationIdentifier/licenseDocumentationIdentifierType	The type of the identifier following a vocabulary. If the identification has been created locally the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
<b>PM108</b>	License documentation identifier value	The identification value of the license documentation.	1..1 MUST

	rights/rightsStatement/licenseInformation/licenseDocumentationIdentifier/licenseDocumentationIdentifierValue		
<b>PM109</b>	Statute information rights/rightsStatement/[rightsBasis='statute'] rights/rightsStatement/statuteInformation	When the rights basis is a statute these elements occur.	0..1 SHOULD
<b>PM110</b>	Statute jurisdiction rights/rightsStatement/statuteInformation/statuteJurisdiction	The country in which the statute is enacted. Use a value from ISO 3166.  See also: <a href="https://www.iso.org/iso-3166-country-codes.html">https://www.iso.org/iso-3166-country-codes.html</a>	1..1 MUST
<b>PM111</b>	Statute citation rights/rightsStatement/statuteInformation/statuteCitation	An identifying designation for the statute.	1..1 MUST
<b>PM112</b>	Statute documentation rights/rightsStatement/statuteInformation/statuteDocumentationIdentifier	It is possible to link to statute documentation to give more information.	0..1 MAY
<b>PM113</b>	Statute documentation identifier type rights/rightsStatement/statuteInformation/statuteDocumentationIdentifier/statuteDocumentationIdentifierType	The type of the identifier following a vocabulary. If the identification has been created locally the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1 MUST
<b>PM114</b>	Statute documentation identifier value rights/rightsStatement/statuteInformation/statuteDocumentationIdentifier/statuteDocumentationIdentifierValue	The identification value of the statute documentation.	1..1 MUST
<b>PM115</b>	Other rights information rights/rightsStatement/[rightsBasis='other']	When rights basis is other these elements occur.	0..1 SHOULD

	rights/rightsStatement/otherRightsInformation		
<b>PM116</b>	Other rights documentation  rights/rightsStatement/otherRightsInformation/otherRightsDocumentationIdentifier	It is possible to link to rights documentation to give more information.	0..1  MAY
<b>PM117</b>	Other rights documentation identifier type  rights/rightsStatement/otherRightsInformation/otherRightsDocumentationIdentifier/otherRightsDocumentationIdentifierType	The type of the identifier following a vocabulary. If the identification has been created locally the value "local" is used.  See also: <a href="https://id.loc.gov/vocabulary/identifiers.html">https://id.loc.gov/vocabulary/identifiers.html</a>	1..1  MUST
<b>PM118</b>	Other rights documentation identifier value  rights/rightsStatement/otherRightsInformation/otherRightsDocumentationIdentifier/otherRightsDocumentationIdentifierValue	The identification value of the other rights documentation.	1..1  MUST
<b>PM119</b>	Other rights basis  rights/rightsStatement/otherRightsInformation/otherRightsBasis	Designation of the basis for the other right or permission described. Should follow a locally created and maintained vocabulary.	1..1  MUST
<b>PM120</b>	Rights Granted  rights/rightsStatement/rightsGranted	Information regarding the rights granted.	0..1  SHOULD
<b>PM121</b>	Acts allowed  rights/rightsStatement/rightsGranted/act	The specific type of act allowed is defined with the vocabulary "Event Type".  See also: <a href="https://id.loc.gov/vocabulary/preservation/eventType.html">https://id.loc.gov/vocabulary/preservation/eventType.html</a>	1..1  MUST
<b>PM122</b>	Dates when the act is allowed  rights/rightsStatement/rightsGranted/termOfGrant	The term for when the grant is applicable.	0..1  SHOULD



<b>PM123</b>	Start date of grant period rights/rightsStatement/rightsGranted/termOfGrant/startDate	The start date for the allowed act.	1..1 MUST
<b>PM124</b>	End date of grant period rights/rightsStatement/rightsGranted/termOfGrant/endDate	The end date for the allowed act.	0..1 MAY
<b>PM125</b>	Note about the rights granted rights/rightsStatement/rightsGranted/rightsGrantedNote	A note regarding the grants (such as a statement about risk assessment when the repository is not certain about the permissions that have been granted).	0..1 MAY

Example of a description of rights.

```
<rights>
  <rightsStatement>
    <rightsStatementIdentifier>
      <rightsStatementIdentifierType>
        local
      </rightsStatementIdentifierType>
      <rightsStatementIdentifierValue>
        discovery-right-001
      </rightsStatementIdentifierValue>
    </rightsStatementIdentifier>
    <rightsBasis>Statute</rightsBasis>
    <rightsGranted>
      <act>displaying</act>
    </rightsGranted>
  </rightsStatement>
</rights>
```

## 6.2 Examples

Two examples are presented in this section. The first example, found in section “6.2.1 Example of representation and its access software”, shows the usual case for presenting in a Dissemination Information Package (DIP) to ensure that the user can open the data object in the DIP. The second example, found in section “6.2.2 Example of a file with an associated agent, event and rights”, is most common in the repository with the Archival Information Package (AIP).

### 6.2.1 Example of representation and its access software

---

```
<?xml version="1.0" encoding="UTF-8"?>
<premis xmlns="http://www.loc.gov/premis/v3"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.loc.gov/premis/v3
http://www.loc.gov/standards/premis/premis.xsd"
  version="3.0">
<!-- A description of the software to use for access -->
  <object xsi:type="intellectualEntity">
    <objectIdentifier>
      <objectIdentifierType>local</objectIdentifierType>
      <objectIdentifierValue>DBPTK</objectIdentifierValue>
    </objectIdentifier>
    <environmentFunction>
      <environmentFunctionType>software</environmentFunctionType>
      <environmentFunctionLevel>1</environmentFunctionLevel>
    </environmentFunction>
    <environmentFunction>
      <environmentFunctionType>
        software application
      </environmentFunctionType>
      <environmentFunctionLevel>2</environmentFunctionLevel>
    </environmentFunction>
    <environmentDesignation>
      <environmentName>DBPTK Desktop</environmentName>
      <environmentVersion>2.5.3</environmentVersion>
      <environmentDesignationNote>
        Desktop application to store database to archival format, validate it and browse the
        content. https://github.com/keeps/dbptk-ui/releases/tag/v2.5.3
      </environmentDesignationNote>
    </environmentDesignation>
  </object>
<!-- A description of the representation, observe that the filepath as identifierType is an
example just used in this specification -->
  <object xsi:type="representation">
    <objectIdentifier>
      <objectIdentifierType>filepath</objectIdentifierType>
      <objectIdentifierValue>
        representations\AVID.SA.18006.rep0
      </objectIdentifierValue>
    </objectIdentifier>
    <significantProperties>
      <significantPropertiesType>
        IP representation format
      </significantPropertiesType>
      <significantPropertiesValue>SIARD2</significantPropertiesValue>
    </significantProperties>
  <!-- A description of the relation between the software and the representation -->
```

```

<relationship>
  <relationshipType>dependency</relationshipType>
  <relationshipSubType>requires</relationshipSubType>
  <relatedObjectIdentifier>
    <relatedObjectIdentifierType>
      Local
    </relatedObjectIdentifierType>
    <relatedObjectIdentifierValue>
      DBPTK
    </relatedObjectIdentifierValue>
  </relatedObjectIdentifier>
  <relatedEnvironmentPurpose>render</relatedEnvironmentPurpose>
</relationship>
</object>
</premis>

```

### 6.2.2 Example of a file with an associated agent, event and rights

```

<?xml version="1.0" encoding="UTF-8"?>
<premis xmlns="http://www.loc.gov/premis/v3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.loc.gov/premis/v3
  http://www.loc.gov/standards/premis/premis.xsd"
  version="3.0">
<!-- A description of a file, observe that the filepath as identifierType is an example just used in
this specification -->
  <object xsi:type="file">
    <objectIdentifier>
      <objectIdentifierType>local</objectIdentifierType>
      <objectIdentifierValue>fileId001</objectIdentifierValue>
    </objectIdentifier>
    <objectIdentifier>
      <objectIdentifierType>filepath</objectIdentifierType>
      <objectIdentifierValue>metadata/file.xml</objectIdentifierValue>
    </objectIdentifier>
    <objectCharacteristics>
      <fixity>
        <messageDigestAlgorithm>SHA-256</messageDigestAlgorithm>
        <messageDigest>3b1d00f7871d9102001c77f...</messageDigest>
        <messageDigestOriginator>/usr/bin/sha256sum</messageDigestOriginator>
      </fixity>
      <format>
        <formatDesignation>
          <formatName>XML</formatName>
          <formatVersion>1.0</formatVersion>
        </formatDesignation>

```

```
<formatRegistry>
  <formatRegistryName>PRONOM</formatRegistryName>
  <formatRegistryKey>fmt/101</formatRegistryKey>
  <formatRegistryRole>specification</formatRegistryRole>
</formatRegistry>
</format>
<objectCharacteristicsExtension>
  <jhove>
    ...
  </jhove>
</objectCharacteristicsExtension>
</objectCharacteristics>
<originalName>originalfilename.ext</originalName>
<storage>
  <contentLocation>
    <contentLocationType>Uniform Resource Identifier</contentLocationType>
    <contentLocationValue>
      /path/to/file.txt
    </contentLocationValue>
  </contentLocation>
  <storageMedium>hard disk HD2253</storageMedium>
</storage>
<relationship>
  <relationshipType>structural</relationshipType>
  <relationshipSubType>is included in</relationshipSubType>
  <relatedObjectIdentifier>
    <relatedObjectIdentifierType>local</relatedObjectIdentifierType>
    <relatedObjectIdentifierValue>
      ID123e4567-e89b-12d3-a456-426655440000
    </relatedObjectIdentifierValue>
  </relatedObjectIdentifier>
  <relatedEventIdentifier>
    <relatedEventIdentifierType>local</relatedEventIdentifierType>
    <relatedEventIdentifierValue>
      ingest-001
    </relatedEventIdentifierValue>
  </relatedEventIdentifier>
</relationship>
<linkingRightsStatementIdentifier>
  <linkingRightsStatementIdentifierType>
    filepath
  </linkingRightsStatementIdentifierType>
  <linkingRightsStatementIdentifierValue>
    metadata/file.xml
  </linkingRightsStatementIdentifierValue>
</linkingRightsStatementIdentifier>
</object>
```

---

```
<event>
  <eventIdentifier>
    <eventIdentifierType>local</eventIdentifierType>
    <eventIdentifierValue>migration-001</eventIdentifierValue>
  </eventIdentifier>
  <eventType>MIGRATION</eventType>
  <eventDateTime>2015-09-01T01:00:00+01:00</eventDateTime>
  <eventOutcomeInformation>
    <eventOutcome>success</eventOutcome>
  </eventOutcomeInformation>
  <linkingAgentIdentifier>
    <linkingAgentIdentifierType>local</linkingAgentIdentifierType>
    <linkingAgentIdentifierValue>
      FileFormatConversion001
    </linkingAgentIdentifierValue>
  </linkingAgentIdentifier>
  <linkingObjectIdentifier>
    <linkingObjectIdentifierType>filepath</linkingObjectIdentifierType>
    <linkingObjectIdentifierValue>
      metadata/file.xml
    </linkingObjectIdentifierValue>
  </linkingObjectIdentifier>
</event>
<agent>
  <agentIdentifier>
    <agentIdentifierType>local</agentIdentifierType>
    <agentIdentifierValue>Indexer</agentIdentifierValue>
  </agentIdentifier>
  <agentName>Indexing Software</agentName>
  <agentType>Software</agentType>
  <linkingRightsStatementIdentifier>
    <linkingRightsStatementIdentifierType>
      local
    </linkingRightsStatementIdentifierType>
    <linkingRightsStatementIdentifierValue>
      discovery-right-001
    </linkingRightsStatementIdentifierValue>
  </linkingRightsStatementIdentifier>
</agent>
<rights>
  <rightsStatement>
    <rightsStatementIdentifier>
      <rightsStatementIdentifierType>
        local
      </rightsStatementIdentifierType>
      <rightsStatementIdentifierValue>
        discovery-right-001
      </rightsStatementIdentifierValue>
    </rightsStatementIdentifier>
  </rightsStatement>
</rights>
```

```
</rightsStatementIdentifierValue>
</rightsStatementIdentifier>
<rightsBasis>Statute</rightsBasis>
<rightsGranted>
  <act>displaying</act>
</rightsGranted>
</rightsStatement>
</rights>
</premis>
```

## 7 Postface

AUTHOR(S)	
Name(s)	Organisation(s)
Karin Bredenberg	Kommunalförbundet Sydarkivera

REVIEWER(S)	
Name(s)	Organisation(s)
Jaime Kaminski	Highbury R&D (Ireland)
Luis Faria	KEEP Solutions LDA

Project co-funded by the European Commission		within the ICT Policy Support Programme
Dissemination Level		
<b>P</b>	<b>Public</b>	X
<b>C</b>	<b>Confidential, only for members of the Consortium and the Commission Services</b>	

## REVISION HISTORY AND STATEMENT OF ORIGINALITY

### Submitted Revisions History

Revision No.	Date	Authors(s)	Organisation	Description
[Version]	[Date]	[Who]	[Affiliation]	[What]
0.1	2020-03-25	Karin Bredenberg	SYD	CITS created
1.0	2021-08-31	Karin Bredenberg	SYD	First publication of CITS
1.0.1	2024-05-17	Karin Bredenberg	SYD	Update of CITS to CS and in PM63 with correct term for obligation

#### **Statement of originality:**

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.