



# RECOMMENDATION FOR external file structure of binary data for the SIARD 2.0 format

**Project Acronym:** E-ARK

**Grant Agreement Number:** 620998

**Project Title:** European Archival Records and Knowledge Preservation

---

AUTHOR(S)	
Name(s)	Organisation(s)
Anders Bo Nielsen	Danish National Archives (DNA)
Kathrine Hougaard Edsen Johansen	Danish National Archives (DNA)
Phillip Tømmerholt	Danish National Archives (DNA)

REVIEWER(S)	
Name(s)	Organisation(s)
Kuldar Aas	National Archives of Estonia (NAE)

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				
Rev. No.	Date	Author(s)	Organisation(s)	Description
0.1	15-05-2015	Anders Bo Nielsen	DNA	
0.5	20-07-2015	Anders Bo Nielsen	DNA	
0.8	15-09-2015	Anders Bo Nielsen	DNA	
0.9	15-10-2015	Anders Bo Nielsen	DNA	
0.10	13-11-2015	Phillip Tømmerholt	DNA	Change of foldername containing dot "." replaced with underscore "_".
0.11	30-11-2015	Anders Bo Nielsen	DNA	Example updated
0.12	8-12-2015	Phillip Tømmerholt	DNA	Example updated
0.13	09-12-2015	Phillip Tømmerholt	DNA	Example updated
0.14	19-01-2016	Phillip Tømmerholt	DNA	Revision
0.15	20-01-2016	Anders Bo Nielsen	DNA	METS example with segmented IP changed
0.16	23-02-2016	Anders Bo Nielsen	DNA	Typos corrected in XML example
0.17	to do...			Change METS example with segmented IP
	to do...			Naming convention for splitting binary files

REVISION HISTORY				
Rev. No.	Date	Author(s)	Organisation(s)	Description
	to do...			Add numbered sections and align styles

## **EXECUTIVE SUMMARY**

A Submission Information Package (SIP) is defined in the OAIS standard<sup>1</sup> as an Information Package that is delivered by the Producer to the OAIS for use in the construction or update of one or more AIPs and/or the associated Descriptive Information. Many different SIP formats are used all over the world and unfortunately there is currently no central format for a SIP which would cover all individual national and business needs identified in the E-ARK Report on Available Best Practices.

---

<sup>1</sup> Reference Model for an Open Archival Information System (OAIS), 2012, [public.ccsds.org/publications/archive/650x0m2.pdf](http://public.ccsds.org/publications/archive/650x0m2.pdf)

## TABLE OF CONTENTS

### Indhold

INTRODUCTION .....	6
Methods for binary data handling in relational databases .....	7
Binary data handling in SQL Standards .....	7
Binary data handling in the SIARD 2.0 format .....	7
Support for internal BLOBS (ISO/IEC 9075-2:2008 - BLOBS) in the SIARD 2.0 format .....	7
Support for external files (ISO/IEC 9075-9:2008 – SQL/MED) in the SIARD 2.0 format .....	7
Recommendation for the folder structure for BLOBS stored outside the SIARD table folder structure .....	9
"Generic" example .....	9
Example with the example database Northwind .....	10
Limits for file amount and for total file size per folder .....	10
Below is an illustration of the folder structure for the example .....	11
Recommendation for using METS to index folders containing BLOBS .....	13
Using METS to refer between child IP(s) and parent IP .....	13
Using METS to refer between parent IP and child IP(s) .....	14
Below is an example for the Northwind database .....	15
Recommendation for the folder structure for external files (SQL/MED) stored outside the SIARD table folder structure .....	19
REFERENCES .....	20
APPENDICIES .....	21
Appendix A: .....	21
Appendix B: Terminology .....	22

## **INTRODUCTION**

This document is a recommendation for the external file structure of binary data for the SIARD 2.0 format. The SIARD 2.0 format has its own specification, which deliberately does not specify the exact external file structure of binary data. This is avoided in order to get a high degree of durability and longevity for the specification, which requires flexibility. Instead, such details are left to recommendations like this one.

This recommendation will be used by the E-ARK open source tool Database Preservation Toolkit (DPT) for the export of relational databases to the SIARD 2.0 format.

## Methods for binary data handling in relational databases

Binary data in regard to relational databases is defined as data for which no simple datatype (such as integer or date) exists. In addition the size of binary data is also important due to efficient handling in databases. Binary data is mostly referred to as binary large object (BLOB). Similarly large amounts of character data are named CLOB, they pose a problem due to size more than lack of a proper data type. For the rest of this recommendations CLOBs will be treated as BLOBs. Databases and handling of binary data has always been a challenge, regardless of the handling was based on:

1. internal BLOBs - where data is contained in the records.
2. external direct references (path and filename) - where BLOBs are stored as files.
3. external indirect reference (file ID)- where BLOBs are stored as files.
4. other methods may exist.

### Binary data handling in SQL Standards

The first method using internal BLOBs has been available for many versions of the SQL standard. It is supported by all current relational database management systems.

The other methods using external files has been available since SQL:2003 and is named Management of External Data (SQL/MED). It is still poorly supported by the current relational database management systems, and perhaps due to lack of detailed specification in the SQL standard those RDBMS that support it do it differently.

### Binary data handling in the SIARD 2.0 format

The SIARD 2.0 format is based on among others standards on the SQL:2008 standard.

#### ***Support for internal BLOBS (ISO/IEC 9075-2:2008 - BLOBS) in the SIARD 2.0 format***

The SIARD 2.0 format specification supports the method in SQL:2008 using internal BLOBS (ISO/IEC 9075-2:2008) as did SIARD 1.0 (SQL:1999).

The SIARD 2.0 format supports BLOBS stored as files inside the SIARD table folder structure and describes this in details in the SIARD 2.0 format specification (similar to SIARD 1.0).

The SIARD 2.0 format supports BLOBS stored as files outside the SIARD table folder structure (new feature in SIARD 2.0), but does not describe this in details, it is left for recommendations like this one.

#### ***Support for external files (ISO/IEC 9075-9:2008 – SQL/MED) in the SIARD 2.0 format***

The SIARD 2.0 format does not support the method in SQL:2008 using external files (ISO/IEC 9075-9:2008 – SQL/MED) due to lack of specification in the SQL standard and especially due to lack of support in the RDBMS. The SIARD 2.0 format does not forbid the use of methods using external files, but leaves it to recommendations like this one to add such support.





## Recommendation for the folder structure for BLOBS stored outside the SIARD table folder structure

This recommendation specifies the folder structure for files stored externally to the SIARD folder structure, i.e. outside of the SIARD ZIP package file.

Due to file systems limitations and media limitations it is currently necessary to limit:

- the amount of files in one folder (such as 10,000 files)
- the size of files in one folder hierarchy (such as 1 TB)

In order to get effective and easy file and folder handling (such as copying, packing, hashing, mounting) new folders are created when reaching a folder amount or folder size limitation, whatever comes first.

These folders are placed at the same top level to get effective and easy file and folder handling.

### "Generic" example

Below is a "generic" example for a single column  $k$  in a single table  $j$ , in which the file amount limit is reached before the file size limit.

[xxx] to be replaced including the [ ]. (very rough BNF)

```
[databaseName].siard <!-- packaged as a ZIP file -->
  content/
  header/
    metadata.xml
    metadata.xsd

[databaseName]_lobseg_[h]/..
  content/schema[i]/table[j]/lob[k]/record[1].bin
  ..
  content/schema[i]/table[j]/lob[k]/record[FILEAMOUNTLIMIT].bin
  <!-- file amount limit per folder reached -->
[databaseName]_lobseg_[h+1]/
  content/schema[i]/table[j]/lob[k]/record[FILEAMOUNTLIMIT+1].bin
  ..
  content/schema[i]/table[j]/lob[k]/record[FILEAMOUNTLIMIT+q].bin
  <!-- total file size limit per folder reached -->
[databaseName]_lobseg_[h+2]/
  content/schema[i]/table[j]/lob[k]/record[FILEAMOUNTLIMIT+q+1].bin
```

Note that the base value for records and tables is 0, whereas it is 1 for columns and row.

## Example with the example database Northwind

Below is an example of the folder structure for the example database Northwind with external LOBs. Northwind includes the following tables which in our example are ordered this way:

Orders (table0),  
Products (table1),  
Categories (table2), - this table contains LOBS which exceed 2000 bytes or 2000 characters.  
Shippers (table3),  
Employees (table4), - this table contains LOBS which exceed 2000 bytes or 2000 characters.  
Territories (table5),  
CustomerDemographics (table6),  
CustomerCostumerDemo (table7),  
Suppliers (table8),  
EmployeeTerritories (table9),  
Customers (table10),  
Sysdiagrams (table11),  
Region (table12),

Only the table "Categories" below is used in this example

CategoryID	CategoryName	Description	Picture
1	Beverages	Soft drinks, coffees, teas, beers, and ales	BLOB (Size: 10151)
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings	BLOB (Size: 12107)
3	Confections	Desserts, candies, and sweet breads	BLOB (Size: 12007)
4	Dairy Products	Cheeses	BLOB (Size: 9756)
5	Grains/Cereals	Breads, crackers, pasta, and cereal	BLOB (Size: 12131)
6	Meat/Poultry	Prepared meats	BLOB (Size: 11280)
7	Produce	Dried fruit and bean curd	BLOB (Size: 12338)
8	Seafood	Seaweed and fish	BLOB (Size: 12069)

### *Limits for file amount and for total file size per folder*

The file amount limit per folder is in this case set to 4 and the total file size limit per folder is set to 45,000 bytes (unrealistic but useful for an example).

Row 1, 2, 3 and 4 will have its LOBs from column 4 ('Picture') (record0.bin, record1.bin, record2.bin, record3.bin) stored together in a folder named Northwind\_lobseg\_0.

Hereby the *file amount limit* of 4 is reached, no more files can be stored in this folder, and therefore a new folder is created named Northwind\_lobseg\_1.

Row 5, 6, and 7 will have its LOBs from column 4 ('Picture') (record4.bin, record5.bin, record6.bin) stored together in a folder named Northwind\_lobseg\_1.

Row 8 will *not* have its LOB from column 4 ('Picture') (record7.bin) stored together with the ones from row 5, 6 and 7 in the folder named Northwind\_lobseg\_1. Not because the *file amount limit* of 4 is reached, but because the *accumulated file size per folder limit* of 45,000 is reached.

The LOBs from row 5, 6 and 7 have respectively a size of 12,131; 11,280 and 12,338 accumulated to 35,749. Adding the LOB from row 8 with a size of 12,069 to the 35,749 of row 5, 6 and 7 would

accumulate to 47,818 and break the *accumulated file size per folder limit* of 45,000. Therefore a new folder is created named Northwind\_lobseg\_2, and Row 8 will have its LOB from column 4 ('Picture') (record7.bin) stored in it.

*Below is an illustration of the folder structure for the example*

Northwind.siard <!-- packaged as a ZIP file --> content/ header/ metadata.xml metadata.xsd
Northwind_lobseg_0/ content/schema0/table2/lob4/record0.bin <!-- row 1, col 4 has record0.bin --> content/schema0/table2/lob4/record1.bin content/schema0/table2/lob4/record2.bin content/schema0/table2/lob4/record3.bin <!-- amount limit reached -->
Northwind_lobseg_1/ content/schema0/table2/lob4/record4.bin content/schema0/table2/lob4/record5.bin content/schema0/table2/lob4/record6.bin <!-- size limit reached -->
Northwind_lobseg_2/ content/schema0/table2/lob4/record7.bin

Below is an extract of metadata.xml using the above folder structure:

<pre> &lt;dbname&gt;Northwind&lt;/dbname&gt; &lt;dataOwner&gt;...&lt;/dataOwner&gt; &lt;dataOriginTimespan&gt;2015&lt;/dataOriginTimespan&gt; &lt;lobFolder&gt;file:///S:/Archives/Northwind/&lt;/lobFolder&gt; </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Note the lobFolder content – it assumes that the Northwind\_lobseg\_[n]/ folders reside in a virtual folder named **Northwind**.<sup>2</sup>

Below is an extract of table2.xml using the above folder structure.

```

<row><c1>2</c1><c2>Condiments</c2><c3>Sweet and savory sauces, relishes, spreads, and
seasonings</c3>
<c4 file="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record1.bin" length="12107"
messageDigest="md522a0cbe8960b78ce48b07a285ce69e3c"/></row>

<row><c1>3</c1><c2>Confections</c2><c3>Desserts, candies, and sweet breads</c3>

```

<sup>2</sup> see <https://tools.ietf.org/html/draft-ietf-appsawg-file-scheme-03#appendix-C.1.1>

```
<c4 file="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record2.bin" length="12007"  
messageDigest="md53e2f2028a9147c29bdcd36ed4e5f25b3"/></row>
```

```
<row><c1>4</c1><c2>Dairy Products</c2><c3>Cheeses</c3>  
<c4 file="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record3.bin" length="9756"  
messageDigest="md512f588040e11cc2021ea37d46aa10c51"/></row>
```

```
<row><c1>5</c1><c2>Grains/Cereals</c2><c3>Breads, crackers, pasta, and cereal</c3>  
<c4 file="file:///Northwind_lobseg_1/content/schema0/table2/lob4/record4.bin" length="12131"  
messageDigest="md5e2d8ef03e1b24edd946820dbbf44fdfd"/></row>
```

```
<row><c1>6</c1><c2>Meat/Poultry</c2><c3>Prepared meats</c3>  
<c4 file="file:///Northwind_lobseg_1/content/schema0/table2/lob4/record5.bin" length="11280"  
messageDigest="md5814a3eb95253c08137f70bcfc279e00f"/></row>
```

```
<row><c1>7</c1><c2>Produce</c2><c3>Dried fruit and bean curd</c3>  
<c4 file="file:///Northwind_lobseg_1/content/schema0/table2/lob4/record6.bin" length="12338"  
messageDigest="md5ee114cd7700f566b1f7c7e8e0f68ca0f"/></row>
```

```
<row><c1>8</c1><c2>Seafood</c2><c3>Seaweed and fish</c3>  
<c4 file="file:///Northwind_lobseg_2/content/schema0/table2/lob4/record7.bin" length="12069"  
messageDigest="md52de1ac4c4e8ebb853e17db01af3fb7c3"/></row>
```

Note that the messageDigest indicates hexadecimal values and it is therefore not of strict importance whether they are set in upper or lower case. However, lower cases are mostly used and enforced, see e.g. RFC 2831 <https://www.ietf.org/rfc/rfc2831.txt>

## Recommendation for using METS to index folders containing BLOBS

In the previous section the recommended folder structure for BLOBs stored outside the SIARD table was specified. This structure allows for the segmentation of large amounts of BLOBs from a database into folders at the same hierarchical level (top level) to get effective and easy file and folder handling. These folders can easily be distributed over many different media to overcome storage and I/O limitations, but will then need an index in order to find these folders.

This section is a recommendation for using METS to index such folders.

Several methods have been described and discussed, and in order to limit the deviation within the E-ARK project we have decided to use the same method which is used for indexing and segmenting IPs in the common specification, which in turn is based on the divided structure of AIPs.

### *Using METS to refer between child IP(s) and parent IP*

The child IP(s) refer to the parent IP using the value of the xlink:href attribute in the <mptr> element. This value must equal the value of the OBJID attribute for the <mets> element in the parent IP mets file. Therefore the same type of ID must be used to these two attributes.

Note that the ID attribute for the <mets> file is for identifying that specific element in the mets file whereas the OBJID attribute in the <mets> element is for identifying the whole mets file (seen as an object.) (according to the mets standard).

Here follows a partial example, where the value of the xlink:href attribute in the <mptr> element (inside the <div> element inside the <structMap> element) is "ID.AVID.RA.18005".

Extract of <mptr> element from child IP mets file:

```
<mets:structMap TYPE="logical" LABEL="parent IP">
  <mets:div LABEL="IP parent identifier">
    <mets:mptr xlink:href="ID.AVID.RA.18005" xlink:title="Referencing the parent IP of this child IP"
    LOCTYPE="OTHER" OTHERLOCTYPE="Some ID scheme" ID="ID755d4d5f-5c5d-4751-9652-fcf839c7c6f2"/>
```

The LOCTYPE attribute has the value "OTHER" and the OTHERLOCTYPE has the value "Some ID scheme".

Instead of "OTHER" for LOCTYPE the value URN, URL, PURL etc. could have been used.

When OTHER is used for LOCTYPE, OTHERLOCTYPE should be used to define the type (and preferably more meaningful than just using "Some ID scheme" in this example).

The value "ID.AVID.RA.18005" must now match the value of the OBJID attribute for the <mets> element in the parent IP mets file.

Extract of mets element from parent IP mets file:

```
mets:mets xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.loc.gov/METS/ schemas/mets.xsd"
OBJID="ID.AVID.RA.18005" ID="ID.AVID.RA.18005.mets.element" TYPE="SIARD2.0 INDEX"
```

**PROFILE**="http://SIARD2.0.xml" **LABEL**="SIARD 2.0 parent">

The parent IP's mets file has the value "ID.AVID.RA.18005" for the attribute OBJID in the <mets> element which matches the value of the xlink:href attribute in the <mptr> element in the child ID's mets file, cf above.

### ***Using METS to refer between parent IP and child IP(s)***

Apart from referring from child IP to parent IP the mets file is also used to refer from parent to child(ren). Thereby not only does the children know its parent, but the parent also knows its children.

The mechanism used to refer from parent to child is by using the value of the xlink:href attribute in the <mptr> element in the mets file of the parent IP.

This value must match the value of the ID attribute of the <file> element in the mets file of the parent IP. The <file> element has a sub element named <FLocat> which has a xlink:href attribute which has the value which is the location of the file for the child IP.

This matching of ID and location is necessary to find the location of the file.

Note that the value of the xlink:href attribute in the <mptr> element for a certain child IP in the mets file of the parent IP is set equal to the value of the OBJID attribute of the <mets> element in the mets file of the child IP. This is sufficient for linking the IDs of the parent and the child, but without the matching value of the ID attribute of the <file> element in the mets file of the parent IP the location of the child IP is not known.

Extract of mptr element from parent IP mets file:

```
<mets:div LABEL="child IPs">
  <mets:div LABEL="child IP">
    <mets:mptr xlink:href="ID_Northwind_lobseg_0_mets" xlink:title="Reference to a child IP"
      LOCTYPE="OTHER" OTHERLOCTYPE="someID" ID="IDd98e416f-55a7-4237-8d45-59c22d221669"/>
```

The xlink:href value "ID\_Northwind\_lobseg\_0\_mets" is equal to the <mets:file> attribute ID value "ID\_Northwind\_lobseg\_0\_mets" in the following:

Extract of file element from parent IP mets file:

```
<mets:fileGrp USE="child IP" ID="IDf98e416f-55a7-4237-8d45-59c22d221663">
  <mets:file ID="ID_Northwind_lobseg_0_mets" MIMETYPE="xml/mets"
    OWNERID="ID.AVID.RA.18005.NORTHWIND" CHECKSUMTYPE="MD5"
    CHECKSUM="c84642ff066720a5b8f4193a7f213782">
    <mets:FLocat LOCTYPE="URL" xlink:type="simple"
      xlink:href="file:///Northwind_lobseg_0/mets.xml"/>
```

The value of the attribute xlink:href is "file:///Northwind\_lobseg\_0/mets.xml"/>".

This child IP has the object

Extract of <mets> element from the child IP whose mets file has the location

Northwind\_lobseg\_0/mets.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<mets:mets xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.loc.gov/METS/
schemas/mets.xsd" OBJID="ID_Northwind_lobseg_0_mets" ID="ID.AVID.RA.18005.seg.0.mets.element"
TYPE="SIARD2.0 INDEX" PROFILE="http://SIARD2.0.xml" LABEL="SIARD 2.0 child IP">
```

The OBJID has the value "ID\_Northwind\_lobseg\_0\_mets"

## Below is an example for the Northwind database

### Parent IP mets file

```
<?xml version="1.0" encoding="UTF-8"?>
<mets:mets xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.loc.gov/METS/schemas/mets.xsd"
OBJID="ID.AVID.RA.18005" ID="ID.AVID.RA.18005.mets.element" TYPE="SIARD2.0 INDEX"
PROFILE="http://SIARD2.0.xml" LABEL="SIARD 2.0 parent">
  <!-- note that the ID and OBJID do not need to be the same - we need to decide what ID should be explicitly
understandable -->
  <mets:metsHdr CREATEDATE="2015-06-13T14:40:00+01:00" RECORDSTATUS="NEW">
    <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
      <mets:name>SIARD 2.0 index</mets:name>
      <mets:note>Created by XX Software following E-ARK recommendation for external LOB
handling in SIARD 2.0</mets:note>
    </mets:agent>
    <mets:metsDocumentID>IP parent SIARD content</mets:metsDocumentID>
  </mets:metsHdr>
  <mets:fileSec>
    <mets:fileGrp ID="ID_SIARD_FILES">
      <!-- the fileGrp for all SIARD files -->
      <mets:fileGrp ID="ID_SIARD_DDL">
        <!-- the fileGrp for the central SIARD file named [databasename].siard -->
        <mets:file ID="ID_NORTHWIND.SIARD" MIMETYPE="OTHER/SIARD"
OWNERID="ID.AVID.RA.18005.NORTHWIND" CHECKSUMTYPE="MD5"
CHECKSUM="A1FB456A233542293459034589034534">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///northwind.siard"/>
          <mets:transformFile TRANSFORMALGORITHM="ZIP64"
TRANSFORMORDER="1" TRANSFORMTYPE="decompression"/>
          <!-- ZIP32 is also allowed for SIARD -->
        </mets:file>
      </mets:fileGrp>
      <mets:fileGrp USE="child IP" ID="IDf98e416f-55a7-4237-8d45-59c22d221663">
        <!-- the fileGrp for the optional external LOBS stored in segments (folders or folders
packaged as files in a package format) at the same parallel level as the SIARD file named .siard -->
        <!-- these files are all referred to in other mets files using the <mtpr> element, therefore only
the mets files ments are here - we need to store the check sum of the mets files in a file as well -->
        <mets:file ID="ID_Northwind_lobseg_0_mets" MIMETYPE="xml/mets"
OWNERID="ID.AVID.RA.18005.NORTHWIND" CHECKSUMTYPE="MD5"
CHECKSUM="c84642ff066720a5b8f4193a7f213782">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_0/mets.xml"/>
        </mets:file>
        <mets:file ID="ID_Northwind_lobseg_1_mets" MIMETYPE="xml/mets"
OWNERID="ID.AVID.RA.18005.NORTHWIND" CHECKSUMTYPE="MD5"
CHECKSUM="132d3d9f7e6a199c893f49b315708c3e">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple"
```

## Project 620998: European Archival Records and Knowledge Preservation - E-ARK

```
xlink:href="file:///Northwind_lobseg_1/mets.xml"/>
    </mets:file>
    <mets:file ID="ID_Northwind_lobseg_2_mets" MIMETYPE="xml/mets"
OWNERID="ID.AVID.RA.18005.NORTHWIND" CHECKSUMTYPE="MD5"
CHECKSUM="03b7eaebbdab6584b9c23dd1895bde21">
    <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_2/mets.xml"/>
    </mets:file>
  </mets:fileGrp>
</mets:fileGrp>
</mets:fileSec>
<mets:structMap LABEL="SIARD structmap" TYPE="PHYSICAL">
  <!-- structmap for the SIARD files - the central SIARD file named x.siard and the possible LOBs as
external files - only one <structmap> element is used for both, even though separate is possible -->
  <mets:div LABEL="SIARD files">
    <mets:div LABEL="SIARD central file">
      <mets:fptr FILEID="ID_NORTHWIND.SIARD"
CONTENTIDS="ID.AVID.RA.18005.Northwind.siard" ID="IDe98e416f-55a7-4237-8d45-59c22d221660"/>
      <!-- the central SIARD file named .siard - packaged as a ZIP64 file -->
    </mets:div>
    <mets:div LABEL="child IPs">
      <mets:div LABEL="child IP">
        <mets:mptr xlink:href="ID_Northwind_lobseg_0_mets" xlink:title="Reference to a
child IP" LOCTYPE="OTHER" OTHERLOCTYPE="someID" ID="IDd98e416f-55a7-4237-8d45-59c22d221669"/>
        <!-- the href value is a link to the mets file -->
      </mets:div>
      <mets:div LABEL="child IP">
        <mets:mptr xlink:href="ID_Northwind_lobseg_1_mets" xlink:title="Reference to a
child IP" LOCTYPE="OTHER" OTHERLOCTYPE="someID" ID="ID70f8ec28-23f1-4364-9163-b3e99165b6e6"/>
      </mets:div>
      <mets:div LABEL="child IP">
        <mets:mptr xlink:href="ID_Northwind_lobseg_2_mets" xlink:title="Reference to a
child IP" LOCTYPE="OTHER" OTHERLOCTYPE="someID" ID="ID3f0cc05c-f27d-499d-a6fd-63bdfed13cb0"/>
      </mets:div>
    </mets:div>
  </mets:div>
</mets:structMap>
</mets:mets>
```



## Child IP mets file

```
<?xml version="1.0" encoding="UTF-8"?>
<mets:mets xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink" xsi:schemaLocation="http://www.loc.gov/METS/ schemas/mets.xsd"
OBJID="ID_Northwind_lobseg_0_mets" ID="ID.AVID.RA.18005.seg.0.mets.element" TYPE="SIARD2.0 INDEX"
PROFILE="http://SIARD2.0.xml" LABEL="SIARD 2.0 child IP">
  <mets:metsHdr CREATEDATE="2015-06-13T14:40:00+01:00" RECORDSTATUS="NEW">
    <mets:agent ROLE="CREATOR" TYPE="OTHER" OTHERTYPE="SOFTWARE">
      <mets:name>SIARD 2.0 index</mets:name>
      <mets:note>Created by XX Software following E-ARK recommendation for external LOB
handling in SIARD 2.0</mets:note>
    </mets:agent>
    <mets:metsDocumentID>IP child SIARD content</mets:metsDocumentID>
  </mets:metsHdr>
  <mets:fileSec>
    <!-- The following fileGrp is for the SIARD files. Other files in this IP must have their own fileGrp -->
    <mets:fileGrp ID="ID_SIARD_FILES">
      <!-- The following fileGrp is for a segment of the SIARD files. Only the parent IP knows the
amount of segments-->
      <mets:fileGrp ID="seg_0">
        <!-- The following OWNERID=table no.row no.column. no -->
        <!-- The following ID is equal to the OWNERID but does not need to be -->
        <mets:file ID="s0.t2.c4.r1" MIMETYPE="binary" OWNERID="s0.t2.c4.r1"
CHECKSUMTYPE="MD5" CHECKSUM="74f24080fc9d234d3ac221b8e743c763" SIZE="10151">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record0.bin"/>
        </mets:file>
        <mets:file ID="s0.t2.c4.r2" MIMETYPE="binary" OWNERID="s0.t2.c4.r2"
CHECKSUMTYPE="MD5" CHECKSUM="22a0cbe8960b78ce48b07a285ce69e3c" SIZE="12107">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record1.bin"/>
        </mets:file>
        <mets:file ID="s0.t2.c4.r3" MIMETYPE="binary" OWNERID="s0.t2.c4.r3"
CHECKSUMTYPE="MD5" CHECKSUM="3e2f2028a9147c29bdcd36ed4e5f25b3" SIZE="12007">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record2.bin"/>
        </mets:file>
        <mets:file ID="s0.t2.c4.r4" MIMETYPE="binary" OWNERID="s0.t2.c4.r4"
CHECKSUMTYPE="MD5" CHECKSUM="12f588040e11cc2021ea37d46aa10c51" SIZE="9756">
          <mets:FLocat LOCTYPE="URL" xlink:type="simple"
xlink:href="file:///Northwind_lobseg_0/content/schema0/table2/lob4/record3.bin"/>
        </mets:file>
      </mets:fileGrp>
    </mets:fileGrp>
  </mets:fileSec>
  <!-- The following structMap is for the files in the segment - design proposal by Karin Bredenberg -->
  <mets:structMap LABEL="SIARD External LOB structmap" TYPE="LOBPOINTER">
    <mets:div LABEL="SIARD LOBs as external files">
      <mets:fptr CONTENTIDS="ID.AVID.RA.18005.seg_0"/>
    </mets:div>
  </mets:structMap>
  <!-- The following structMap is to inform that this child IP belongs to a certain parent ID using the value of the
```

## Project 620998: European Archival Records and Knowledge Preservation - E-ARK

xlink:href attribute which must match the value of the OBJID attribute for the <mets> element in the parent IP mets file ; design proposal by AIT -->

```
<mets:structMap TYPE="logical" LABEL="parent IP">
  <mets:div LABEL="IP parent identifier">
    <mets:mptr xlink:href="ID.AVID.RA.18005" xlink:title="Referencing the parent IP of this child
IP" LOCTYPE="OTHER" OTHERLOCTYPE="Some ID scheme" ID="ID755d4d5f-5c5d-4751-9652-fcf839c7c6f2"/>
  </mets:div>
</mets:structMap>
</mets:mets>
```

**Recommendation for the folder structure for external files  
(SQL/MED) stored outside the SIARD table folder structure**

## REFERENCES

1. A Checklist for Documenting PREMIS-METS Decisions in a METS Profile, 2010, URL: [http://www.loc.gov/standards/premis/premis\\_mets\\_checklist.pdf](http://www.loc.gov/standards/premis/premis_mets_checklist.pdf)
2. Database Preservation Toolkit, 2015, URL: <http://keeps.github.io/db-preservation-toolkit/>
3. Classified information, 2015, URL: [http://en.wikipedia.org/wiki/Classified\\_information](http://en.wikipedia.org/wiki/Classified_information)
4. E-ARK Report on Available Best Practices, 2014, URL: <http://eak-project.com/resources/project-deliverables/6-d31-e-ark-report-on-available-best-practices>
5. e-SENS (Electronic Simple European Networked Services) project, 2015, URL: <http://www.esens.eu/>
6. Encoded Archival Context for Corporate Bodies, Persons, and Families, 2015, URL: <http://eac.staatsbibliothek-berlin.de/>
7. Executive\_Order\_on\_Submission\_Information\_Packages, 2010, URL: [http://www.sa.dk/media%283367,1033%29/Executive\\_Order\\_on\\_Submission\\_Information\\_Packages.pdf](http://www.sa.dk/media%283367,1033%29/Executive_Order_on_Submission_Information_Packages.pdf)
8. FGS packet structure, 2013, URL: [https://riksarkivet.se/Media/pdf-filer/Projekt/FGS\\_Earkiv\\_Paket.pdf](https://riksarkivet.se/Media/pdf-filer/Projekt/FGS_Earkiv_Paket.pdf)
9. Guidelines for using PREMIS with METS for exchange, Revised September 17, 2008 URL: <http://www.loc.gov/standards/premis/guidelines-premismets.pdf>.
10. Media Types, 2015, URL: <https://www.iana.org/assignments/media-types/media-types.xhtml>
11. METS, 2015, URL: <http://www.loc.gov/standards/mets/>
12. METS Profile Components, 2011, URL: [http://www.loc.gov/standards/mets/profile\\_docs/components.html](http://www.loc.gov/standards/mets/profile_docs/components.html)
13. METS Profiles, 2012, URL: <http://www.loc.gov/standards/mets/mets-profiles.html>
14. Planets project, 2007, URL: <http://www.planets-project.eu/>
15. Producer, Submission Agreements: Glossary of Terms, 2015, URL: <http://sites.tufts.edu/dca/about-us/research-initiatives/taper-tufts-accessioning-program-for-electronic-records/project-documentation/submission-agreements-glossary-of-terms/>
16. Producer-Archive Interface Methodology Abstract Standard (PAIMAS), 2004, URL: [public.ccsds.org/publications/archive/651x0m1.pdf](http://public.ccsds.org/publications/archive/651x0m1.pdf)
17. Producer-Archive Interface Specification (PAIS) – CCSDS, 2014, URL: [public.ccsds.org/publications/archive/651x1b1.pdf](http://public.ccsds.org/publications/archive/651x1b1.pdf)
18. Records Creator, Submission Agreements: Glossary of Terms, 2015, URL: <http://sites.tufts.edu/dca/about-us/research-initiatives/taper-tufts-accessioning-program-for-electronic-records/project-documentation/submission-agreements-glossary-of-terms/>
19. Reference Model for an Open Archival Information System (OAIS), 2012, URL: [public.ccsds.org/publications/archive/650x0m2.pdf](http://public.ccsds.org/publications/archive/650x0m2.pdf)
20. SIARD Specification, 2015, URL: <http://www.bar.admin.ch/dienstleistungen/00823/01911/index.html>

## **APPENDICIES**

### **Appendix A:**

## Appendix B: Terminology

<b>Archive*</b>	An Organisation that intends to preserve information for Access and use by a Designated Community.
<b>Content type</b>	The delivery-specific content submission type.
<b>ERMS</b>	Is a type of content management system and refers to the electronic records management system.
<b>Information Package*</b>	A logical container composed of optional Content Information and optional associated Preservation Description Information. Associated with this Information Package is Packaging Information used to delimit and identify the Content Information and Package Description information used to facilitate searches for the Content Information.
<b>Ingest Functional Entity*</b>	The OAIS functional entity that contains the services and functions that accept Submission Information Packages from Producers, prepares Archival Information Packages for storage, and ensures that Archival Information Packages and their supporting Descriptive Information become established within the OAIS.
<b>OAIS*</b>	The Open Archival Information System is an archive (and a standard: ISO 14721:2003), consisting of an organisation of people and systems that has accepted the responsibility to preserve information and make it available for a Designated Community.
<b>Submission Information Package (SIP)*</b>	An Information Package that is delivered by the Producer to the OAIS for use in the construction or update of one or more AIPs and/or the associated Descriptive Information.
<b>Submitting organisation</b>	Name of the organisation submitting the package to the archive. Extends the delivery information since it may be the case that the content of a creator is held by another part of the organisation.
<b>Delivering organisation</b>	The organisation delivering the package to the archive. For stating and extending the information use of element "Producer organisation name" and "Submitting organisation name" is recommended.
<b>Archival creator<sup>3</sup></b>	An organization unit or individual that creates records and/or manages

<sup>3</sup> Records Creator, Submission Agreements: Glossary of Terms, 2015, <http://sites.tufts.edu/dca/about-us/research-initiatives/taper-tufts-accessioning-program-for-electronic-records/project-documentation/submission-agreements>

	those records during their active use.
<b>Producing organisation<sup>4</sup></b>	<p>The organizational unit or individual that has the authority to transfer records to an archive. Usually the producer is also the records creator, the organizational unit or individual that created and managed the records during their active use.</p> <p>This is not always the case, sometimes the producer is different from the records creator.</p> <p>For example: An author dies and her literary executor gains the authority to transfer her papers to an archive. The author is the records creator and the literary executor is the producer.</p> <p>For example: Department X gets reorganized out of existence and Department Y, which takes over the functional responsibilities of Department X, gains the authority to transfer the records of Department X to the archive. Department X is the records creator and Department Y is the producer.</p> <p>Counter example: The Department of Widget Science transfers some of its own records to the archive. The Department of Widget Science is the records creator and the producer.</p>

\* Reference Model for an Open Archival Information System (OAIS), 2012, <http://public.ccsds.org/publications/archive/650x0m2.pdf>