ECP-2008-DILI-538013

MIMO

Metadata Mapping and OAI-PMH Implementation Guidelines

Deliverable number	*D-2.2*
Dissemination level	*Restricted*
Delivery date	*18 November 2010*
Status	*v1.0*
Author(s)	*Thierry Le Meur, Rodolphe Bailly*

e*Content*plus

This project is funded under the *e*Content*plus* programme[1],
a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

[1] OJ L 79, 24.3.2005, p. 1.

1 SCOPE OF THE DOCUMENT 5
METADATA MAPPING GUIDELINES 6
2 METADATA MAPPING PROCESS 7
3 WHICH FIELD TO WHICH LIDO ELEMENT? 7
3.1 MANDATORY INFORMATION 7
3.2 OPTIONAL INFORMATION 8
4 RECORD IDENTIFICATION (MANDATORY) 10
5 LANGUAGE SPECIFICATION (MANDATORY) 11
6 INSTRUMENT CLASSIFICATION (MANDATORY) 12
6.1 OBJECT TYPE (MANDATORY) 12
6.2 INSTRUMENT CLASSIFICATION (MANDATORY) 13
7 INSTRUMENT IDENTIFICATION (MANDATORY) 13
7.1 TITLE (MANDATORY) 13
7.2 INSCRIPTIONS (OPTIONAL) 14
7.3 REPOSITORY (OPTIONAL) 15
7.3.1 Inventory number (optional) 15
7.3.2 Repository name (mandatory) 15
7.4 INSTRUMENT EDITION (OPTIONAL) 17
7.5 INSTRUMENT DESCRIPTION (OPTIONAL) 17
7.6 MEASUREMENTS (OPTIONAL) 19
8 EVENT DESCRIPTION (OPTIONAL) 20
8.1 EVENT GLOBAL DESCRIPTION (OPTIONAL) 20
8.2 EVENT IDENTIFICATION (OPTIONAL) 21
8.3 EVENT TYPE (MANDATORY) 21
8.4 EVENT NAME (OPTIONAL) 21
9 EVENT ACTOR (OPTIONAL) 22
9.1 ACTOR AND ROLE GLOBAL DESCRIPTION (OPTIONAL) 22
9.2 ACTOR TYPE (MANDATORY) 22
9.3 ACTOR IDENTIFIER (OPTIONAL) 22
9.4 ACTOR NAME (MANDATORY) 23
9.5 ACTOR NATIONALITY (OPTIONAL) 24
9.6 ACTOR VITAL DATES (OPTIONAL) 24
9.7 ACTOR GENDER (OPTIONAL) 25
9.8 ACTOR ROLE (OPTIONAL) 25
10 EVENT CULTURE (OPTIONAL) 26
11 EVENT DATES (OPTIONAL) 26
12 EVENT PLACE (OPTIONAL) 28
12.1 PLACE GLOBAL DESCRIPTION (OPTIONAL) 28
12.2 PLACE IDENTIFIER (OPTIONAL) 28
12.3 PLACE NAME (OPTIONAL) 28
12.4 LIDO AND GEONAMES 29
13 EVENT MATERIALS AND TECHNIQUES 35
14 SUBJECT (OPTIONAL) 35
15 RIGHTS (OPTIONAL) 35
15.1 RIGHTS TYPE (MANDATORY) 35
15.2 RIGHTS DATE (OPTIONAL) 36
15.3 RIGHTS HOLDER (OPTIONAL) 36

15.4 CREDIT LINE (OPTIONAL) 37
16 RECORD (MANDATORY) 37
16.1 RECORD IDENTIFIER (MANDATORY) 37
16.2 RECORD TYPE (MANDATORY) 38
16.3 RECORD SOURCE (MANDATORY) 38
16.4 LINK TO THE RECORD (OPTIONAL) 39
17 RESOURCE (OPTIONAL) 39
17.1 RESOURCE IDENTIFIER (OPTIONAL) 39
17.2 LINK TO THE RESOURCE (OPTIONAL) 40
17.3 RESOURCE TYPE (OPTIONAL) 40
OAI-PMH IMPLEMENTATION GUIDELINES 41
18 BASIC FUNCTIONING OF OAI-PMH 42
19 OAI-PMH IMPLEMENTION TECHNIQUES 43
19.1 "DYNAMIC" REPOSITORY 43
19.2 "PHYSICAL" REPOSITORY 44
19.3 STATIC REPOSITORY 45
20 OVERVIEW OF OAI-PMH REQUESTS AND RESPONSES 46
20.1 IDENTIFY 46
20.2 LISTMETADATAFORMATS 48
20.3 LISTIDENTIFIERS 50
20.4 LISTRECORDS 51
20.5 GETRECORD 53
21 STATIC REPOSITORY XML FILE EXAMPLE 55
22 OAI-PMH 2.0 SPECIFICATION AND IMPLEMENTATION DOCUMENTS 58
23 OAI-PMH 2.0 TOOLKITS 58

Document history

Date		Version	Author
2010/01/27	Document creation	v0.1	Thierry Le Meur
2010/07/08	§ 12.4 LIDO and GeoNames clarify the way event places have to be described in LIDO model and gives some recommendations to facilitate the best retrieval of these event places in GeoNames.	v0.2	Thierry Le Meur
2010/11/18	new version 1.0 of LIDO schema	v1.0	Thierry Le Meur

1 Scope of the document

This document provides detailed specifications for the development of metadata repositories to be harvested by the MIMO aggregator.
Using OAI-PMH, the aggregator will harvest metadata describing musical instruments from data providers. Data providers expose their metadata into repositories. LIDO v1.0 is the format of the metadata available through these repositories (see MIMO D2.1).
Therefore, data providers must convert their data into LIDO using mapping rules between their format and LIDO.

The first part of this document describes mapping rules to be applied between a generic museum data model and LIDO.
The second part emphasizes the different techniques to implement OAI repositories.

MIMO
musical instrument museums online

Metadata mapping guidelines

2 Metadata mapping process

The following table describes the mapping process for each metadata element

Contributor's metadata element	LIDO element
value of the contributor's metadata element →	<lido:element> value of the LIDO element </lido: element>

In general, the mapping process consists in simply copying the value of a metadata element available on the contributor's database to the corresponding lido:element.

Mapping can also be a way to enrich the metadata (for example by providing information such as the language of the metadata). Such information is usually not included into the original record. It is added on the fly during the mapping process.

3 Which field to which LIDO element?

3.1 Mandatory information

The following table gives the metadata elements that must be provided when defining a LIDO record.

Metadata element	LIDO element
identifier of the record (on MIMO platform): *composed of the identifier of the original record in the local contributor's local system and an identifier for the contributor.*	lidoRecID
language used to write the descriptive metadata: *MIMO values = "de, "en", "fr", "it", "nl", "sv".*	xml:lang attribute of lido:descriptiveMetadata
object type: *MIMO values = "musical instrument", "parts of musical instruments".*	lido:objectWorkType
instrument classification: *the keyword that designates in a unique way the instrument in MIMO's instruments list.*	lido:classification
name of the instrument: *the name or title given to the instrument.*	lido:titleSet
inventory number of the instrument: *the current inventory number gave to the instrument by its institution of custody.*	lido:workID
repository where the instrument is kept: *the name of the institution of custody.*	lido:repositoryName

language used to write the administrative metadata: *MIMO values = "de, "en", "fr", "it", "nl", "se".*	xml:lang attribute of lido:administrativeMetadata
original identifier of the record: *the identifier of the record describing the metadata on the contributor's local system.*	lido:recordID
collection level of the record *MIMO values = "item".*	lido:recordType
identification of the institution managing the metadata (generally the institution of custody itself): *the identifier of the institution*	lido:recordSource

3.2 Optional information

The following tables give the metadata elements that may be provided when defining a LIDO record.

Information about the instrument itself

Contributor's metadata element	LIDO element
information about the inscriptions: *a text describing or giving a transcription of any mark made on the instrument.*	lido:inscriptions
information about the instrument edition: *in general, the serial number in case of a industrially manufactured instrument.*	lido:displayStateEdition
descriptions: *texts giving a general and/ or specific descriptions of the instrument.*	lido:objectDescriptionSet
measurements: *dimensions or measurements of the instrument or of a part of the instrument.*	lido:measurementsSet

Information about the history of the instrument (events description)

Contributor's metadata element	LIDO element
event type: *qualifier of an event chosen into MIMO's event types list.* ***This information is the only one which is mandatory when describing an event.***	lido:eventType
event name: *a text naming or describing the event.*	lido:eventName
actor(s) involved in the event: *information (identity, dates, nationality, role, etc…) about persons or firms which participated in the event.*	lido:eventActor

the name of an actor may be taken from the MIMO's authority list of instrument makers.	
culture *the keyword that designates in a unique way a people in MIMO's ethnonyms list.*	lido:culture
the date(s) of the event: *when did the event take place.*	lido:eventDate
the place of the event: *a place name taken from MIMO's geographical list.*	lido:eventPlace
material(s) and/or techniques *what was used for making, modifying, repairing… the instrument.*	lido:eventMaterialTech

Information about the subjects

A subject is what is represented by an instrument decorative element such as drawing, painting or engraving.

Contributor's metadata element	LIDO element
the description of the subject: *a text giving this description.*	lido:displaySubject

Information about the rights related to the metadata

Contributor's metadata element	LIDO element
the nature of the rights attached to the metadata: *"copyrights", "trademark", etc…*	lido:rightsType
the period during which the rights holder held the rights.	lido:rightsDate
the name of the rights holder (person or institution)	lido:rightsHolder
acknowledgement of the rights associated with the metadata.	lido:creditLine

Information about the metadata themselves

Contributor's metadata element	LIDO element
the URL of the page on the contributor's website displaying the metadata.	lido:recordInfoLink

Information about the resources attached to the instrument

A resource is a digital document (image, movie, sound recording...) showing an instrument or its use.

Contributor's metadata element	LIDO element
the name of the file containing the resource on the contributor's information system.	lido:resourceID
URL to the resource.	lido:linkResource
the type of the resource: *MIMO values = "image", "movie", "sound"…*	lido:resourceType
the name of the institution/person holding the rights attached to the resource.	lido:rightsResource
resource creation or last modification date.	lido:resourceDateTaken

The following paragraphs (§4 to §17) describe the metadata mapping process for each element. The mandatory or optional character of the element is always mentioned in the title of the paragraph.

Within an element, sub-elements themselves may be mandatory or optional: this characteristic is also mentioned in the table.

If a sub-element is mandatory in an optional element, this means that the sub-element must be defined when defining the element.

4 Record identification (mandatory)

Contributor's metadata element	LIDO element
record identifier →	<lido:lidoRecID lido:type=”local”> <contributor identifier>":"<record identifier> </lido:lidoRecID>

Each record must have a unique identifier on the contributor’s local system. This element must be mapped to a lido:lidoRecID element.

This mapped identifier should be built using the following pattern:

<contributor identifier>":"<record identifier>

where <record identifier> is the identifier of the record in the contributor's s local system and <contributor identifier> a prefix that identify the institution from which the record originates.

Each institution participating to the MIMO project has to determine its own contributor identifier. We suggest using the participant short names:

University of Edinburgh	UEDIN
Germanisches Nationalmuseum, Nürnberg	GNM
University of Leipzig, Museum für Musikinstrumente	ULEI
Africamuseum, Tervuren	RMCA

Associazione "Amici del Museo degli Strumenti Musicali", Firenze	AF
Università degli Studi di Firenze , Florence	UF
Cité de la Musique, Paris	CM
Musical Instrument Museum, Brussels	MIM-BE
Horniman Museum, London	HML
Stiftung Preußischer Kulturbesitz, Ethnologisches Museum	SPK
The Stockholm Music Museum	SMS-MM

Example:

```
<lido:lidoRecID lido:type="local">
        CM:0162260
</lido:lidoRecID>
```

In this example, the identifier is composed of two elements, the prefix CM which stands for "Cité de la Musique", and a numeric string, "0162260", which is the identifier of the notice from which information is taken to build the record.
The string giving the identifier of the record in the contributor's s local system may be alphanumeric but should not contain the ":" character.

5 Language specification (mandatory)

Contributor's metadata element	LIDO element
contributor has to specify: a language code for descriptive metadata and for administrative metadata	<lido:descriptiveMetadata xml:lang=language code> … </lido:descriptiveMetadata> <lido:administrativeMetadata xml:lang=language code> … </lido:administrativeMetadata>

Contributors must specifies the main language used to write the descriptive and administrative metadata in the xml:lang attribute attached to lido:descriptiveMetadata element and to lido:administrativeMetadata element.
Values for this attribute must be one of the followings:

"de" for German
"en" for English
"fr" for French
"it" for Italian
"nl" for Dutch
"sv" for Swedish

This list will be completed as other institutions using different languages join the project.

Example 1:

<lido:descriptiveMetadata xml:lang="de">

Example 2:

<lido:administrativeMetadata xml:lang="sv">

Important remark about language information
Europeana will provide multilingual search through its portal. This is one of their biggest technical challenges. One of the requirements to help implement such a search may be that each field of each record must specify the language of the information contained in the field. Therefore, if the information is not present, they may have to assume that the language of the fields is the same as the one of the record itself. Even worse, if no language is provided at the record level, they'll probably use the spoken language of the contributor.
Taking this into account, we heavily encourage data providers to specify themselves the language of the elements especially if they know that this language is different from the language specified at an upper level.

6 Instrument classification (mandatory)

6.1 Object type (mandatory)

Contributor's metadata element	LIDO element
contributor has to determine whether the described object belongs to the set of: "musical instruments" "parts of musical instruments"	<lido:objectWorkType> <lido:term> the chosen set </lido:term> </lido:objectWorkType>

This mandatory information specifies the specific kind of object being described and has to be mapped to a lido:objectWorkType element.
In MIMO context, there are only two types available:
"musical instruments"
"parts of musical instruments"

Example1:

<lido:objectWorkType>
<lido:term>
musical instruments
</lido:term>
</lido:objectWorkType>

Example 2:

```
<lido:objectWorkType>
    <lido:term>
        parts of musical instruments
    </lido:term>
</lido:objectWorkType>
```

6.2 Instrument classification (mandatory)

Contributor's metadata element	LIDO element
The contributor has to provide at least one of the MIMO's intrument keywords. →	<lido:classification> <lido:term> the keyword </lido:term> </lido:classification>

Example1:

```
<lido:classification>
    <lido:term>
        Arghul
    </lido:term>
</lido:classification>
```

Example2:

```
<lido:classification>
    <lido:term>
        Accordion
    </lido:term>
</lido:classification>
```

The model allows specifying more than one classification term.

7 Instrument identification (mandatory)

7.1 Title (mandatory)

Contributor's metadata element	LIDO element
title of the instrument (mandatory) →	<lido:titleSet> <lido:appellationValue> (mandatory) the title of the instrument

source giving this title(optional) →	</lido:appellationValue> <lido:sourceAppellation> (optional) the source </lido:sourceAppellation> </lido:titleSet>

Example 1:

```
<lido:titleSet>
    <lido:appellationValue>
        Violon "le Tua"
    </lido:appellationValue>
</lido:titleSet>
```

Example 2:

```
<lido:titleSet>
    <lido:appellationValue>
        Harpe arquée "kundi"
    </lido:appellationValue>
</lido:titleSet>
```

If several titles are specified for one instrument, the lido:titleSet element has to be repeated and the lido:pref attribute of lido:appellationValue element has to be set to "preferred" for the main title and to "alternate" for the others.

7.2 Inscriptions (optional)

Contributor's metadata element	LIDO element
description of any incription present on the instrument →	<lido:inscriptions> <lido:inscriptionDescription> <lido:descriptiveNoteValue> description of the inscription <lido:descriptiveNoteValue> <lido:inscriptionDescription> </lido:inscriptions>

Example:

```
<lido:inscriptions>
    <lido:inscriptionDescription>
        <lido:descriptiveNoteValue>
            Antonio Stradivarius Cremonensis / Faciebat Anno 1708
        </lido:descriptiveNoteValue>
    </lido:inscriptionDescription>
</lido:inscriptions>
```

If several inscriptions are attached to an instrument the lido:inscriptions element has to be repeated for each inscription inside the lido:inscriptionsWrap.

The location of the inscription may be described into the lido:descriptiveNoteValue.

7.3 Repository (optional)

7.3.1 Inventory number (optional)

Contributor's metadata element	LIDO element
the unambiguous numeric or alphanumeric identification number (inventory number) assigned to the instrument by its institution of custody →	<lido:workID type="inventory number"> the inventory number </lido:workID>

Example:

```
<lido:workID type="inventory number">
        E.979.2.1
</lido:workID>
```

7.3.2 Repository name (mandatory)

Contributor's metadata element	LIDO element
repository identifier if available (optional) →	<lido:repositoryName> <lido:legalBodyID lido:type="local"> (optional) *repository identifier* </lido:legalBodyID> <lido:legalBodyName> (mandatory) <lido:appellationValue> repository name (mandatory) → *repository name* </lido:appellationValue> </lido:legalBodyName> <lido:legalBodyWeblink> (optional) URL of the repository web site (optional) → *URL of the repository web site* </lido:legalBodyWeblink> </lido:repositoryName>

A repository is a legal body. The identification of a legal body is given by the following information:

its identifier
its name
the URL of its web site

At least, the name has to be specified.

Example:

```
<lido:repositoryName>
        <lido:legalBodyName>
                <lido:appellationValue>
                        Musée de la musique
                </lido:appellationValue>
        </lido:legalBodyName>
</lido:repositoryName>
```

workID and repositoryName elements are then linked together into a repositorySet element.

Example:

```
<lido:repositorySet>
        <lido:workID type="inventory number">
                E.979.2.1
        </lido:workID>
        <lido:repositoryName>
                <lido:legalBodyName>
                        <lido:appellationValue>
                                Musée de la musique
                        </lido:appellationValue>
                </lido:legalBodyName>
        </lido:repositoryName>
</lido:repositoySet>
```

The lido:repositoryType attribute of a lido:repositoySet element may be set to "current" or "former" depending on the fact that the repository name is the current name of the repository or a former one.

Example:

```
<lido:repositorySet type="current">
        <lido:workID type="inventory number">
                E.979.2.1
        </lido:workID>
        <lido:repositoryName>
                <lido:legalBodyName>
                        <lido:appellationValue>
                                Musée de la musique
                        </lido:appellationValue>
```

```
                    </lido:legalBodyName>
                </lido:repositoryName>
            </lido:repositoySet>
            <lido:repositorySet type="former">
                <lido:repositoryName>
                    <lido:legalBodyName>
                        <lido:appellationValue>
                            Collection Geneviève Thibaud de Chambure
                        </lido:appellationValue>
                    </lido:legalBodyName>
                </lido:repositoryName>
            </lido:repositoySet>
```

7.4 Instrument edition (optional)

Contributor's metadata element	LIDO element
the free text describing information like serial number, etc… →	<lido:displayEdition> the free text describing the information </lido:displayEdition>

This information applies to instruments industrially manufactured. It can be a serial number, for instance.

Example:

```
<lido:displayEdition>
    serial number : 80360
</lido:displayEdition>
```

7.5 Instrument description (optional)

Contributor's metadata element	LIDO element
the type of description (optional) → *MIMO values = see below*	<lido:objectDescriptionSet lido:type=the chosen type>
the text of the description (mandatory) →	<lido:descriptiveNoteValue> (mandatory) the text of the description </lido:descriptiveNoteValue>
the source giving this description(optional) →	<lido:sourceDescriptiveNote> (optional) the source </lido:sourceDescriptiveNote>
	</lido:objectDescriptionSet>

The description of an instrument is given by:
- the descriptive notes about the instrument
- the source from which the content of the notes comes (optional)

The lido:objectDescriptionSet contains 2 sub-elements to which the previous elements of description can be mapped:
- lido:descriptiveNoteValue
- lido:sourceDescriptiveNote

The lido:objectDescriptionSet can be repeated as many times as there are descriptive notes.
In MIMO's, context, the different lido:objectDescriptionSet should be typed (using lido:type) using the following values:
- "general description"
- "context" (e.g. description of the cultural context of the use of an instrument)
- "decorative features"
- "literature references"
- "performance characteristics" (e.g. advices or comments given by the players)
- "inscriptions locations"

If no type is specified, the description has to be considered as a general description.

Example 1:

```
<lido:objectDescriptionSet>
    <lido:descriptiveNoteValue type="general description">
        Fond en 1 pièce d'érable…
    </lido:descriptiveNoteValue>
    <lido:sourceDescriptiveNote>
        Musée instrumental du CNSM : La facture instrumentale européenne, 1985
    </lido:sourceDescriptiveNote>
</lido:objectDescriptionSet>
```

Example 2:

```
<lido:objectDescriptionSet type="context">
    <lido:descriptiveNoteValue>
        description of the cultural context in which a non-European instrument is used
    </lido:descriptiveNoteValue>
</lido:objectDescriptionSet>
```

7.6 Measurements (optional)

Contributor's metadata element	LIDO element
a free text describing the measurements of the instrument →	<lido:displayObjectMeasurements> the text describing the measurements </lido:displayObjectMeasurements>
measurement split into elementary information: - a text specifying what is measured →	<lido:measurementsSet> <lido:measurementType> what is measured .....</lido:measurementType>
- the unit in which the measurement is given →	<lido:measurementUnit> the unit of the measurement .....</lido:measurementUnit>
- the value of the measurement →	.....<lido:measurementValue> value of the measurement .....</lido:measurementValue> </lido:measurementsSet>

A measure is described:

by a free text describing it in a readable form

and/or

by separate specifications of:

unit (m, cm, mm…)

"dimension" (length, width, height…)

value

If a measurement is described by a free text, it needs to be mapped to a lido:displayObjectMeasurement element.

If a measurement is split into type, unit and value, it needs to be mapped to a lido:measurementsSet element which contains 3 sub-elements:

lido:measurementType

lido:measurementUnit

lido:measurementValue

When used, these 3 sub-elements are mandatory.

All the measurements of an instrument need to be grouped together in one or several lido:objectMeasurements elements.

Example:

```
<lido:objectMeasurementWrap>
    <lido:objectMeasurementSet>
        <lido:displayObjectMeasurements>
            Total length: 2250 mm - Keyboard width: 815 mm
        </lido:displayObjectMeasurements>
        <lido:objectMeasurements>
            <lido:measurementsSet>
```

```
                    <lido:measurementType>
                        Total length
                    </lido:measurementType>
                    <lido:measurementUnit>
                        mm
                    </lido:measurementUnit>
                    <lido:measurementValue>
                        2250
                    </lido:measurementValue>
                </lido:measurementsSet>
                <lido:measurementsSet
                    <lido:measurementType>
                        Keyboard width
                    </lido:measurementType>
                    <lido:measurementUnit>
                        mm
                    </lido:measurementUnit>
                    <lido:measurementValue>
                        815
                    </lido:measurementValue>
                </lido:measurementsSet>
            </lido:objectMeasurements>
        </lido:objectMeasurementSet>
    </lido:objectMeasurementWrap>
```

8 Event description (optional)

8.1 Event global description (optional)

Contributor's metadata element	LIDO element
a free text globally describing the event →	<lido:displayEvent> the text describing the event </lido:displayEvent>

Example:

```
    <lido:displayEvent>
        Ravalement: Pascal Joseph Taskin (1723 - 1793), facteur
    </lido:displayEvent>
```

8.2 Event identification (optional)

Contributor's metadata element	LIDO element
event identifier →	<lido:eventID lido:type="local"> <event identifier> </lido:eventID>

Direct copy in the lido:eventID element of the UID (Unique Identifier) of event in the data provider database.

8.3 Event type (mandatory)

Contributor's metadata element	LIDO element
the contributor has to choose the most accurate type in MIMO's event types list →	<lido:eventType> event type </lido:eventType>

Contributors must specify a type for each event. This information has to be mapped to an lido:eventType element. Values for this attribute must be one of the followings:
of specific importance are:

Acquisition, Creation, Finding, Modification, Use.

Further event types include:

Collecting, Designing, Destruction, Excavation, Exhibition, Loss, Move, Order, Part addition, Part removal, Performance, Planning, Production, Provenance, Publication, Restoration, Transformation, Type assignment, Type creation.

8.4 Event name (optional)

Contributor's metadata element	LIDO element
the name of the event (mandatory) →	<lido:eventName> <lido:appellationValue> (mandatory) the name of the event </lido:appellationValue>
the source giving this name (optional) →	<lido:sourceAppellation> (optional) the source </lido:sourceAppellation> </lido:eventName>

9 Event actor (optional)

9.1 Actor and role global description (optional)

Contributor's metadata element	LIDO element
a free text globally describing the actor and its role in the event →	<lido:displayActorInRole> the text describing the actor and its role </lido:displayActorInRole>

Example:

```
<lido:displayActorInRole>
        Andreas Ruckers (1607-1655), facteur
</lido:displayActorInRole>
```

9.2 Actor type (mandatory)

Contributor's metadata element	LIDO element
for each actor, contributor has to choose its type between the two following values: "person" "firm" →	<lido:actor lido:type=the chosen type>
	</lido:actor>

Example:

```
<lido:actor lido:type="person">
        <lido:nameActorSet>
                <lido:appellationValue>
                        Ruckers, Andreas
                </lido:appellationValue>
        </lido:nameActorSet>
</lido:actor>
```

9.3 Actor identifier (optional)

Contributor's metadata element	LIDO element
actor identifier →	<lido:actorID> <actor identifier> </lido:actorID>

Direct copy in the lido:actorID element of the UID (Unique Identifier) of event in the data provider database.

9.4 Actor name (mandatory)

Contributor's metadata element	LIDO element
the name of the actor (mandatory) →	<lido:actorNameSet> <lido:appellationValue> (mandatory) the actor name </lido:appellationValue>
the source giving this name (optional) →	<lido:sourceAppellation> (optional) the source </lido:sourceAppellation> </lido:actorNameSet>

The name of an actor may be found in the MIMO's authority list of instrument makers. For an individual, his/her name may be composed of a last name and a forename. If so, the value stored in the lido:actorNameSet element is the concatenation of the last name and of the forename separated by the string ", ".

Example 1:

```
<lido:nameActorSet>
        <lido:appellationValue>
                Pleyel
        </lido:appellationValue>
</lido:nameActorSet>
```

Example 2:

```
<lido:nameActorSet>
        <lido:appellationValue>
                Ruckers, Andreas
        </lido:appellationValue>
</lido:nameActorSet>
```

If an actor is known under several names, the lido:actorNameSet element must be repeated for each name and the lido:pref attribute of lido:actorNameSet element has to be set to "preferred" for the main name and to "alternate" for the others.

9.5 Actor nationality (optional)

Contributor's metadata element	LIDO element
actor nationality →	<lido:nationalityActor> <lido:term> actor nationality </lido:term> </lido:nationalityActor>

Example:

```
<lido:nationalityActor>
    <lido:term>
        French
    </lido:term>
</lido:nationalityActor>
```

9.6 Actor vital dates (optional)

Contributor's metadata element	LIDO element
actor's birth date (person) or actor's beginning of activity (firm) →	<lido:vitalDatesActor> <lido:earliestDate> actor's birth date/activity beginning </lido:earliestDate>
actor's death date (person) or actor's end of activity (firm) →	<lido:latestDate> actor's death date/activity end </lido:latestDate> </lido:vitalDatesActor>

The dates format should be preferably YYYY[-MM[-DD]]: at least the year should be specified. No other formats are accepted (e.g 17??, 1850 ?, ca 1600, etc…).
A date may be typed to indicate if it is an exact date or an estimated one.

Example:

```
<lido:vitalDatesActor>
    <lido:earliestDate lido:type="estimatedDate">
    1567
    </lido:earliestDate>
    <lido:earliestDate lido:type="exactDate">
    1643-11-29
    </lido:latestDate>
</lido:vitalDatesActor>
```

9.7 Actor gender (optional)

Contributor's metadata element	LIDO element
actor gender → *MIMO's values = see below*	<lido:genderActor> actor gender </lido:genderActor>

The different values for this element are:

"male"
"female"
"unknown"
"not applicable" (for actors which are not individuals)

Example:

```
<lido:genderActor>
        male
</lido:genderActor>
```

9.8 Actor role (optional)

Contributor's metadata element	LIDO element
actor role →	<lido:roleActor> <lido:term> actor role </lido:term> </lido:roleActor>

The contributor maps directly the information he owns about the actor role into the lido:roleActor element.

Example:

```
<lido:roleActor>
        <lido:term>
                Instrument maker
        </lido:term>
</lido:roleActor>
```

A lido:roleActor element is associated to an lido:actor element within a lido:actorInRole element.

Example:

```
<lido:actorInRole>
        <lido:actor lido:type="person">
                <lido:nameActorSet>
                        <lido:appellationValue>
```

```
                                    Ruckers, Andreas
                                </lido:appellationValue>
                        </lido:nameActorSet>
                </lido:actor>
                <lido:roleActor>
                        <lido:term>
                                Instrument maker
                        </lido:term>
                </lido:roleActor>
        </lido:actorInRole>
```

10 Event culture (optional)

Contributor's metadata element	LIDO element
the people name →	<lido:culture> <lido:term> the people name </lido:term> </lido:culture>

Example:

```
        <lido:culture>
                <lido:term>
                        Mangbetu
                </lido:term>
        </lido:culture>
```

Data contributors should use the people names available in MIMO's ethnonyms list to be provided by WP3.

11 Event dates (optional)

Contributor's metadata element	LIDO element
a free text globally describing the event date (optional) →	<lido:eventDate> <lido:displayDate> (optional) the text describing the date </lido:displayDate> …..<lido:date> …..…..<lido:earliestDate> (mandatory)

the date of beginning of the event (mandatory) →	…..…….the earliest date …..…..</lido:earliestDate> …..…..<lido:latestDate> (mandatory)
the date of ending of the event (mandatory) →	…..…….the latest date …..…..</lido:latestDate> …..…..<lido:periodName> (optional)
the name of the period during which the event happened (optional) →	…..…….the period name …..…..</lido:periodName> …..</lido:date> </lido:eventDate>

Event date is/are described by:

by a free text describing it in a readable form

and/or

by separate specifications of:

the date of beginning

the date of end

the period name

If an event date is described by a free text, it has to be mapped to a lido:displayDate element.

If a date is split into beginning, end and period name, it has to be mapped to a lido:date element which contains 3 sub-elements:

lido:earliestDate

lido:latestDate

lido:periodName

If the date of the event is a precise date the latest date must have the same value than the earliest date.

The dates format should be preferably YYYY[-MM[-DD]]: at least the year should be specified. Nevertheless, other formats are accepted (e.g 17??, 1850 ?, ca 1600, etc…).

Example:

```
<lido:eventDate>
    <lido:date>
        <lido:earliestDate>
            1637
        </lido:earliestDate>
        <lido:latestDate>
            1715
        </lido:latestDate>
    </lido:date>
</lido:eventDate>
```

12 Event place (optional)

12.1 Place global description (optional)

Contributor's metadata element	LIDO element
a free text globally describing the event place →	<lido:displayPlace> the text describing the place </lido:displayPlace>

12.2 Place identifier (optional)

Contributor's metadata element	LIDO element
place identifier →	<lido:placeID> <place identifier> </lido:placeID>

Direct copy in the lido:placeID element of the UID (Unique Identifier) of place in the data provider database.

12.3 Place name (optional)

Contributor's metadata element	LIDO element
the place name (mandatory) →	<lido:placeNameSet> <lido:appellationValue> (mandatory) the place name </lido:appellationValue>
the source giving this name (optional) →	<lido:sourceAppellation> (optional) the source </lido:sourceAppellation> </lido:placeNameSet>

If a place has several names, the lido:placeNameSet element must be repeated for each place name and the lido:pref attribute of lido:placeNameSet element has to be set to "preferred" for the main name and to "alternate" for the others.

12.4 LIDO and GeoNames

Place specification in LIDO model

According to LIDO model, an event place is described by a structure (lido:eventPlace element) composed of, at least, one and only one lido:place element describing the information about the geographical location of the event. The minimal information to know about a place is its name. The specification of this name can be made using the LIDO element called lido:namePlaceSet.

The following example shows how to specify a place named "Paris" using LIDO model:

```
<lido:eventPlace>
   <lido:place>
      <lido:namePlaceSet>
         <lido:appellationValue>
            Paris
         </lido:appellationValue>
      </lido:namePlaceSet>
   </lido:place>
</lido:eventPlace>
```

Places and parts of place

Question: in the previous, how can we make the difference between Paris, capital of France and Paris, small town in Texas?

A place itself can be part of a bigger geographical area. LIDO model allows to describe those combination using the lido:partOfPlace element. The structure of the lido:partOfPlace element is the same as the structure of the lido:place element: the minimal information to know about a part of place is its name. The specification of this name can be made using the lido:namePlaceSet element. A partOfPlace may also be included in another partOfPlace.

Thus, to precise which "Paris" we are talking about, we can say if this place "Paris" is a part of "France" or of "Texas" using the lido:partOfPlace element:

```
<lido:place>
   <lido:namePlaceSet>
      <lido:appellationValue>
         Paris
      </lido:appellationValue>
   </lido:namePlaceSet>
   <lido:partOfPlace>
      <lido:namePlaceSet>
         <lido:appellationValue>
            France
         </lido:appellationValue>
```

```
<lido:place>
   <lido:namePlaceSet>
      <lido:appellationValue>
         Paris
      </lido:appellationValue>
   </lido:namePlaceSet>
   <lido:partOfPlace>
      <lido:namePlaceSet>
         <lido:appellationValue>
            Texas
         </lido:appellationValue>
```

```
        </lido:namePlaceSet>
    </lido:partOfPlace>
</lido:place>
```

```
        </lido:namePlaceSet>
    </lido:partOfPlace>
</lido:place>
```

If we want to be yet more precise, we can specify that "France" is a part of "Europe" and "Texas" a part of "United States of America" themselves part of "North America".

```
<lido:place>
    <lido:namePlaceSet>
        <lido:appellationValue>
            Paris
        </lido:appellationValue>
    </lido:namePlaceSet>
    <lido:partOfPlace>
        <lido:namePlaceSet>
            <lido:appellationValue>
                France
            </lido:appellationValue>
        </lido:namePlaceSet>
        <lido:partOfPlace>
            <lido:namePlaceSet>
                <lido:appellationValue>
                    Europe
                </lido:appellationValue>
            </lido:namePlaceSet>
        </lido:partOfPlace>
    </lido:partOfPlace>
</lido:place>
```

```
<lido:place>
    <lido:namePlaceSet>
        <lido:appellationValue>
            Paris
        </lido:appellationValue>
    </lido:namePlaceSet>
    <lido:partOfPlace>
        <lido:namePlaceSet>
            <lido:appellationValue>
                Texas
            </lido:appellationValue>
        </lido:namePlaceSet>
        <lido:partOfPlace>
            <lido:namePlaceSet>
                <lido:appellationValue>
                    United States of America
                </lido:appellationValue>
            </lido:namePlaceSet>
            <lido:partOfPlace>
                <lido:namePlaceSet>
                    <lido:appellationValue>
                        North America
                    </lido:appellationValue>
                </lido:namePlaceSet>
            </lido:partOfPlace>
        </lido:partOfPlace>
    </lido:partOfPlace>
</lido:place>
```

Places types

The previous examples seem to show the way to resolve any confusion between homonyms. But, in same cases, it's not sufficient. If we consider the following place description:

```
<lido:place>
    <lido:namePlaceSet>
        <lido:appellationValue>
            Mayenne
        </lido:appellationValue>
    </lido:namePlaceSet>
    <lido:partOfPlace>
```

```
        <lido:namePlaceSet>
            <lido:appellationValue>
                France
            </lido:appellationValue>
        </lido:namePlaceSet>
        <lido:partOfPlace>
            <lido:namePlaceSet>
                <lido:appellationValue>
                    Europe
                </lido:appellationValue>
            </lido:namePlaceSet>
        </lido:partOfPlace>
    </lido:partOfPlace>
</lido:place>
```

"Mayenne" may be either a town, a "department" (administrative region) or even a river.

LIDO model allows giving a type to a place (or to a part of place). This can be done by using the attributes lido:politicalEntity and/or geographicalEntity.
lido:politicalEntity refers to political or administrative organization of a territory ;
lido:geographicalEntity refers to any other type of geographical characteristic.

LIDO doesn't specify any list of values for these attributes.

For instance, we can imagine that one value for the lido:politicalEntity could be "country" and so describe the place "France" as follows:

```
<lido:place lido:politicalEntity="country">
    <lido:namePlaceSet>
        <lido:appellationValue>
            France
        </lido:appellationValue>
    </lido:namePlaceSet>
</lido:place>
```

Same thing for "Europe" which can be described as a continent by setting the lido:geographicalEntity with this value "continent":

```
<lido:place lido:politicalEntity="continent">
    <lido:namePlaceSet>
        <lido:appellationValue>
            Europe
        </lido:appellationValue>
    </lido:namePlaceSet>
</lido:place>
```

And if "Mayenne" designates the eponymous town, we have the following description:

```
<lido:place lido:politicalEntity="town">
   <lido:namePlaceSet>
      <lido:appellationValue>
         Mayenne
      </lido:appellationValue>
   </lido:namePlaceSet>
   <lido:partOfPlace lido:politicalEntity="country">
      <lido:namePlaceSet>
         <lido:appellationValue>
            France
         </lido:appellationValue>
      </lido:namePlaceSet>
      <lido:partOfPlace lido:geographicalEntity="continent">
         <lido:namePlaceSet>
            <lido:appellationValue>
               Europe
            </lido:appellationValue>
         </lido:namePlaceSet>
      </lido:partOfPlace>
   </lido:partOfPlace>
</lido:place>
```

GeoNames

Considering the way locations are described in LIDO model and after having experimented GeoNames web services (especially GeoNames web service search), it is possible to give some recommendations in order to:

- specify a location in the simpliest possible manner with LIDO model
- interrogate GeoNames with the best chance to obtain the most relevant answer

1 – When possible, specify the type of each location by setting the one of the two attributes: politicalEntity OR the geographicalEntity

According to the tests we've done with our own records, the following list of values seems to be sufficient to resolve most of ambiguities that can occur.

List of values for politicalEntity attribute

- "**country**" (a political entity)
- "**region**" (always refers to an administrative division of a country)
- "**town**" (from the smallest villages to the major cities)

List of values for geographicalEntity attribute

- "**continent**"
- "**area**" (any location which is not a continent, country, region or a town: an island, a desert, a stream, a geographical area – Middle-East, for instance - …)

These values refer to the way GeoNames categorizes geographical objects and manages continents, countries and what it calls "feature classes" (those interested may have a look to this page: http://www.geonames.org/export/codes.html).

2 – When possible, specify as precisely as possible the hierarchy to which a location belongs by using combination of partOfPlace elements

If the location refers to a continent or a country, the specification of the continent or the country is sufficient.
When the location is an area, specify the country and/or the continent.
When the location is a region, specify at least the country.
When the location is a town, specify at least the country (and, if necessary, the region).

3 – Specify the language in which the name of the location is expressed by setting the xml:lang attribute which is associated to the appellation value of the lido:namePlaceSet element

Example:

```
<lido:place lido:geographicalEntity="area">
    <lido:namePlaceSet>
        <lido:appellationValue xml:lang="de">
            Balearische Inseln
        </lido:appellationValue>
    </lido:namePlaceSet>
    <lido:partOfPlace lido:politicalEntity="region">
        <lido:namePlaceSet>
            <lido:appellationValue xml:lang="de">
                Katalonien
            </lido:appellationValue>
        </lido:namePlaceSet>
        <lido:partOfPlace lido:politicalEntity="country">
            <lido:namePlaceSet>
                <lido:appellationValue xml:lang="de">
                    Spanien
                </lido:appellationValue>
            </lido:namePlaceSet>
            <lido:partOfPlace lido:geographicalEntity="continent">
                <lido:namePlaceSet>
                    <lido:appellationValue xml:lang="de">
                        Europa
                    </lido:appellationValue>
                </lido:namePlaceSet>
            </lido:partOfPlace>
        </lido:partOfPlace>
    </lido:partOfPlace>
</lido:place>
```

Remarks

1 – List of continents as managed by GeoNames

- Africa
- Antartica
- Asia
- Europe
- Oceania
- North America
- South America

GeoNames doesn't know the generic term "America": it uses only the two terms "North America" and "South America" (even the term "Central America" is not defined).

"South America" covers the following countries:

- Argentina
- Bolivia
- Brazil
- Chile
- Colombia
- Ecuador
- Falkland Islands (considered as a country)
- French Guiana (considered as a country)
- Guyana
- Peru
- Paraguay
- Suriname
- Uruguay
- Venezuela

All other American countries depend from "North America" including all the Caribbean islands.

2 – List of countries

Some dependant territories (for instance British or French overseas territories) are considered as countries (see complete list of countries).

13 Event materials and techniques

Contributor's metadata element	LIDO element
type of the term *MIMO's values = "material" or "technique"* the term naming the material or the technique	<lido:termMaterialsTech lido:type=the chosen type> <lido:term> the term naming the material or the technique </lido:term> </lido:actor>

Example:

```
<lido:termMaterialsTech lido:type="material">
        <lido:term>
                Ivory
        </lido:term>
</lido:termMaterialsTech>
```

14 Subject (optional)

Contributor's metadata element	LIDO element
text describing a subject matter represented on an instrument	<lido:displaySubject> the text describing the subject </lido:displaySubject>

15 Rights (optional)

15.1 Rights type (mandatory)

Contributor's metadata element	LIDO element
rights type specification ("copyrights", "trademark",…)	<lido:rightsType> <lido:term> rights type specification </lido:term> </lido:rightsType>

Example:

```
<lido:rightsType>
        <lido:term>
                copyrights
        </lido:term>
</lido:rightsType>
```

15.2 Rights date (optional)

Contributor's metadata element	LIDO element
to specify the dates attached to rights, a contributor has to map at least the two first dates (begin and end):	<lido:rightsDate>
	…..<lido:earliestDate> (mandatory)
the date of beginning of the rights (mandatory) →	……the earliest date
	…..</lido:earliestDate>
	…..<lido:latestDate> (mandatory)
the date of ending of the rights (mandatory) →	……the latest date
	…..</lido:latestDate>
	…..<lido:periodName> (optional)
the name of the period during which the rights are current (optional) →	……the period name
	…..</lido: periodName>
	</lido:rightsDate>

15.3 Rights holder (optional)

Contributor's metadata element	LIDO element
	<lido:rightsHolder>
	<lido:legalBodyID lido:type="local"> (optional)
rights holder identifier (optional) →	*rights holder identifier*
	</lido:legalBodyID>
	<lido:legalBodyName> (mandatory)
	<lido:appellationValue>
rights holder name (mandatory) →	*rights holder name*
	</lido:appellationValue>
	</lido:legalBodyName>
	<lido:legalBodyWeblink> (optional)
URL of the rights holder web site (optional) →	*URL of the right holder web site*
	</lido:legalBodyWeblink>
	</lido:rightsHolder>

Same process as for repository name (see Repository name)

Example:

```
<lido:rightsHolder>
        <lido:legalBodyName>
                <lido:appellationValue>
                        Foto Marburg
                </lido:appellationValue>
        </lido:legalBodyName>
        <lido:legalBodyWeblink>
                http://www.fotomarburg.de
        </lido:legalBodyWeblink>
</lido:rightsHolder>
```

15.4 Credit line (optional)

Contributor's metadata element	LIDO element
credit line →	<lido:creditLine> credit line </lido:creditLine>

For instance, a credit line may refer to the photographer who took the picture or may specify any acknowledgement related to an authorization of using an image, etc…

Example:

```
<lido:creditLine>
        © Agence Photographique
</lido:creditLine>
```

16 Record (mandatory)

16.1 Record identifier (mandatory)

Contributor's metadata element	LIDO element
record identifier →	<lido:recordID lido:type=”local”> record identifier as it is on the contributor's local system </lido:recordID>

This element (lido:recordID) contains the of the record in the contributor's database.

Example:

```
<lido:recordID lido:type="local">
        0162260
</lido:recordID>
```

16.2 Record type (mandatory)

Contributor's metadata element	LIDO element
only one possible value: "item" →	<lido:recordType> <lido:term> item </lido:term> </lido:recordType>

16.3 Record source (mandatory)

Contributor's metadata element	LIDO element
source identifier (mandatory) →	<lido:rightsHolder> <lido:legalBodyID lido:type="local"> (mandatory) *source identifier* </lido:legalBodyID>
source name (optional) →	<lido:legalBodyName> (optional) <lido:appellationValue> *source name* </lido:appellationValue> </lido:legalBodyName>
URL of the source web site (optional) →	<lido:legalBodyWeblink> (optional) *URL of the source web site* </lido:legalBodyWeblink> </lido:rightsHolder>

Same process as for repository name (see Repository name)

Example:

```
<lido:recordSource>
        <lido:legalBodyID lido:type="local">
                CM
        </lido:legalBodyID>
</lido:recordSource>
```

16.4 Link to the record (optional)

Contributor's metadata element	LIDO element
URL of a page displaying the metadata →	<lido:recordInfoSet> <lido:recordInfoLink> URL </lido:recordInfoLink> </lido:recordInfoSet>

This element contains a URI to a web page on the contributor's web site. This web page usually display the metadata contained in the record, and gives access to the digital resources associated to the instrument. This is equivalent to Europeana "isShownBy" element.

Example:

```
<lido:recordInfoSet>
    <lido:recordInfoLink>
        http://mediatheque.cite-musique.fr/masc/play.asp?ID=0162260
    </lido:recordInfoLink>
</lido:recordInfoSet>
```

17 Resource (optional)

17.1 Resource identifier (optional)

Contributor's metadata element	LIDO element
the identifier of the resource on the contributor's local system →	<lido:resourceID lido:type="local"> the identifier of the resource </lido:resourceID>

The resource identifier is generally the name of the file on the contributor's local system.

Example:

```
<lido:resourceID lido:type="local">
    CMIM000015238.jpg
</lido:resourceID>
```

17.2 Link to the resource (optional)

Contributor's metadata element	LIDO element
the URL of the page where the resource can be seen or heard →	<lido:resourceRepresentation> <lido:linkResource> URL of the resource </lido:linkResource> </lido:resourceRepresentation>

This element contains a URI to the resource file itself.
This is equivalent to Europeana "isShownAt" element.

Example:

```
<lido:linkResource>
        http://id.gnm.de/img/MI85_1279807329777_2/jpg
</lido:linkResource>
```

17.3 Resource type (optional)

Contributor's metadata element	LIDO element
the type of the resource on the contributor's local system: "text", "image", "sound", "video" →	<lido:resourceType> <lido:term> the type of the resource </lido:term> </lido:resourceType>

Example:

```
<lido:resourceType>
        image
</lido:resourceType>
```

MIMO
musical instrument museums online

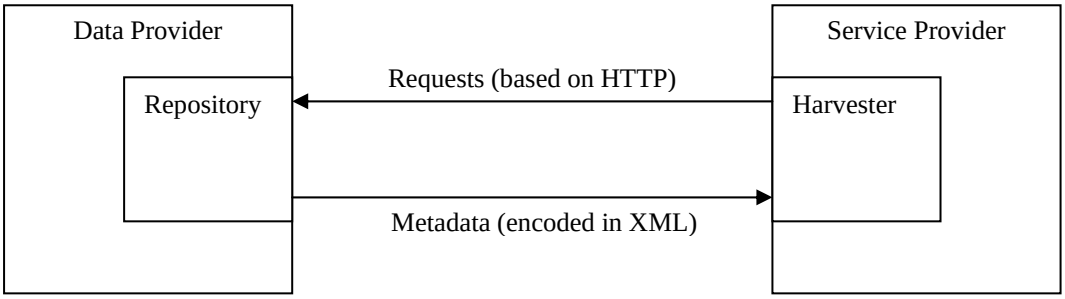
OAI-PMH implementation guidelines

18 Basic functioning of OAI-PMH

The OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) enables effective and efficient sharing of digital metadata and is being utilized across a wide spectrum of disciplines and digital library projects. It is currently in its second version (launched in2002) and is a relatively mature and well-established standard.
Its success is due to the simplicity of its principles and to its flexibility: for example, notions like sets allow harvesting of precise subsets of a database; a data provider may expose its metadata to different harvesters with different needs. Implementation of OAI-PMH on a Museum Collection Management System can be the starting point of multiple automated data exchange with other information systems located in other Institution, or even in the same institution as the Museum, such as the institutional web site.

Note: A correct implementation of an OAI-PMH repository requires the data to be available at least using Dublin Core format (see http://dublincore.org/). In the context of MIMO, LIDO will be mandatory, not Dublin core. Anyway, in order to be fully compatible with the growing community of open data providers, we strongly encourage MIMO contributors to expose their data using Dublin Core, in addition to LIDO.

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is a mechanism for repository interoperability. Data Providers are repositories that expose XML structured metadata via OAI-PMH. Service Providers then make OAI-PMH service requests to harvest that metadata.

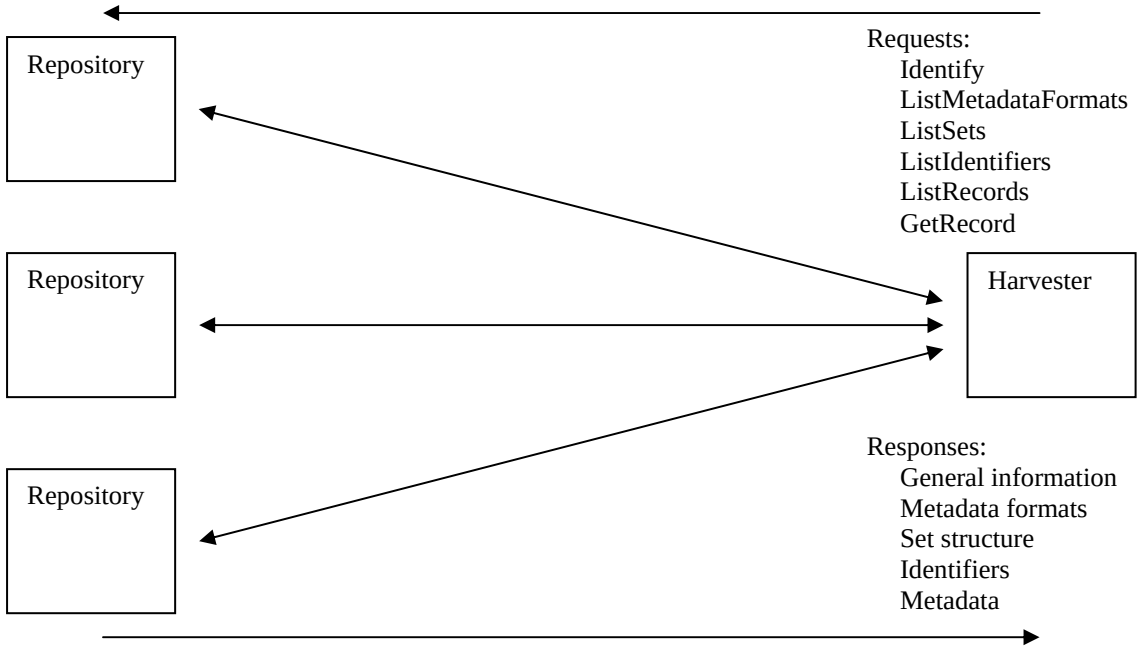


OAI-PMH is a set of six verbs or services that are invoked within HTTP:

- Identify
- ListMetadataFormats
- ListSets
- ListIdentifiers
- ListRecords
- GetRecord

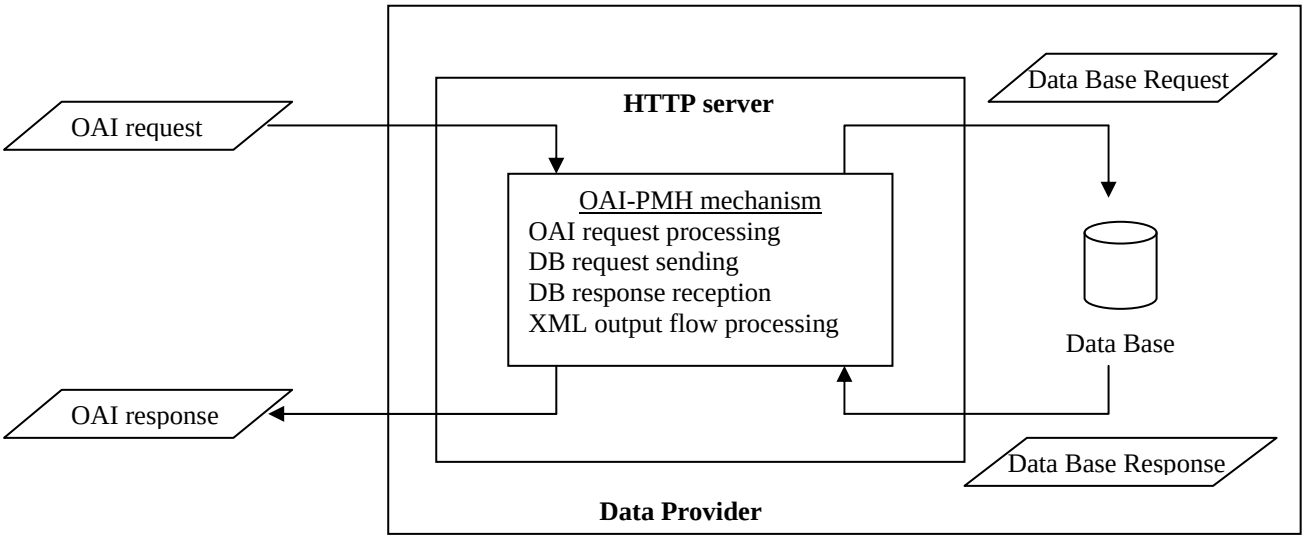
From the responses sent by repositories harvesters extract:

information about repositories (URL, name, administrator email(s), technical information)
metadata formats supported by repositories
sets (sub repositories) structure
lists of records identifiers
metadata themselves



19 OAI-PMH implemention techniques

19.1 "Dynamic" repository



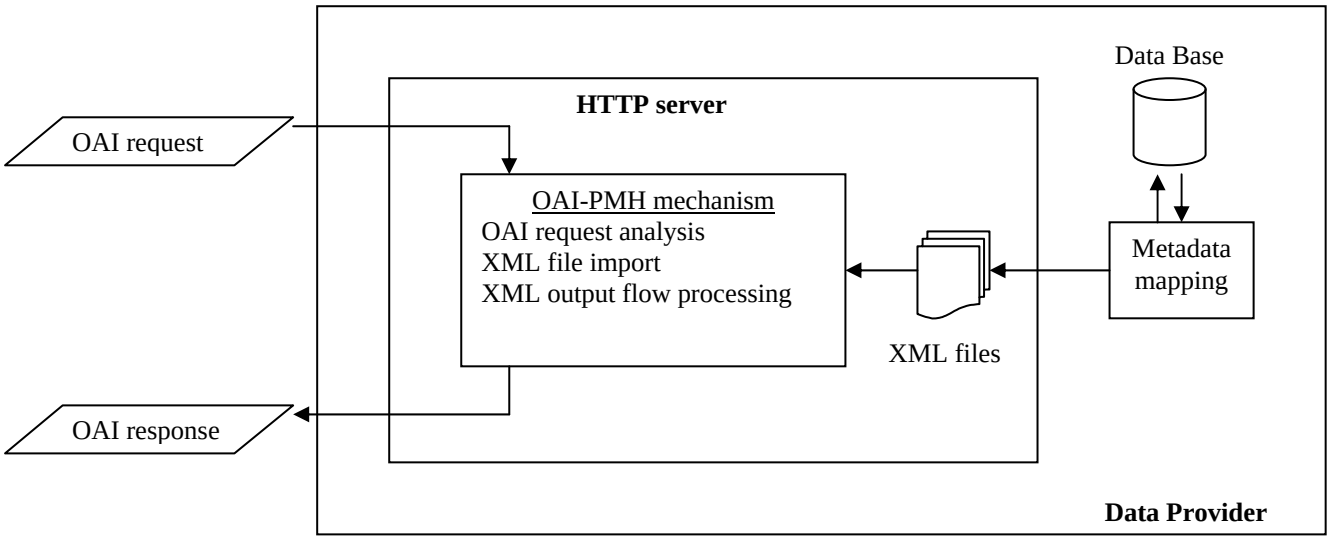
The OAI-PMH mechanism is implemented within the HTTP server and manages the following functions:

- parsing of OAI requests and rejection of badly formed ones
- retrieval of the relevant data from the database according to a request arguments
- generation of the OAI responses directly into the relevant XML scheme

The OAI-PMH mechanism is able to transform "on-the-fly" the data received from the data base into a XML flow describing the metadata using the right data model.
The main advantage of this approach is that the data available through the repository is always up to date.

Open source tools are available to provide the infrastructure which can be very useful to shorten the development time (in particular the implementation of the protocol, which in turn lowers the total cost of implementation. See OAI-PMH 2.0 toolkits).
Using such a tool, the development will be restrained to the transformation (on the fly) of PMH requests into database requests, and to the conversion of these request responses into LIDO records embedded into OAI records.

19.2 "Physical" repository



In this case, the process is split in two parts:

- the metadata mapping from the local data base into one or several XML files
- the OAI-PMH mechanism which only manages the exchange protocol

This scenario is simpler than the previous one, as there is less effort required to set up the OAI mechanism. The focus will be in setting up the metadata mapping process.

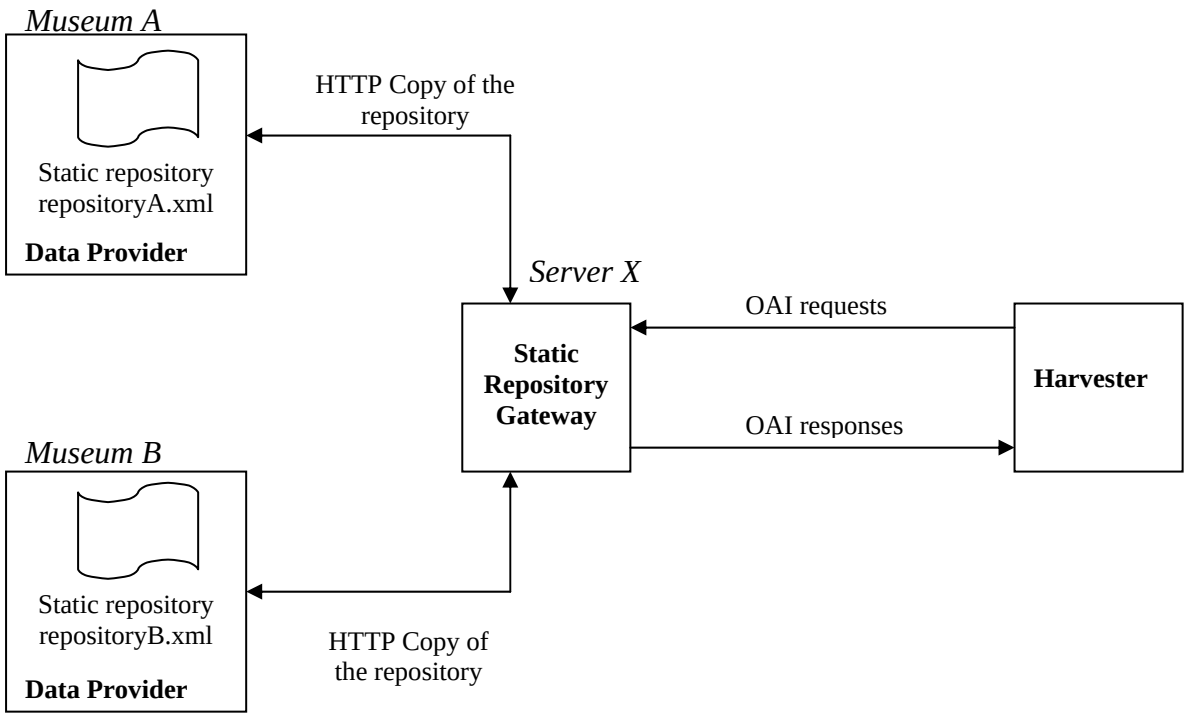
19.3 Static repository

A static repository is a unique XML file, available at a persistent HTTP URL, containing metadata records and the information required for the purpose of harvesting via the OAI-PMH through the intermediation of a static repository gateway.
A static repository is not an OAI-PMH Repository, because it is a file, not a server that can respond to the six OAI-PMH requests.

Static repositories are useful for small and relatively static metadata collections. All information is managed in a single XML file. This file may be created manually with an XML editing tool, or a text processing application. Alternatively, a static repository might be generated periodically by a script that extracts information from an existing database.

A static repository gateway provides intermediation for one or more static repositories. It assigns each static repository a unique static repository base URL, all with a common static repository gateway URL prefix, and thereby exposes each individual static repository as an individual OAI-PMH repository.

Example:



Museum A exposes a static repository which URL is
http://www.MuseumA.org/repositoryA.xml

Museum B exposes a static repository which URL is
http://www.MuseumB.org/repositoryB.xml

The static repository gateway URL is
http://gateway.WebsiteX.org/oai

The harvester will act as if it addresses 2 different repositories of which URLs are
http://gateway.WebsiteX.org/www.MuseumA.org/repositoryA.xml and
http://gateway.WebsiteX.org/www.MuseumB.org/repositoryB.xml

An example of a static repository in given paragraph "Static repository XML file example" MIMO will implement a static repository gateway. Therefore, data providers may use this method to provide metadata to MIMO's platform.

20 Overview of OAI-PMH requests and responses

OAI-PMH requests must be submitted using either the HTTP GET or POST methods.
In addition to the base URL, all requests consist of a list of keyword arguments, which take the form of key=value pairs. Arguments may appear in any order and multiple arguments must be separated by ampersands [&].
Each OAI-PMH request must have at least one key=value pair that specifies the OAI-PMH request issued by the harvester:

- key is the string 'verb'
- value is one of the defined OAI-PMH requests.

The following paragraphs give a quick overview of the OAI-PMH requests in their simplest form without going into more complex technical points like the management of sets and resumption tokens.

20.1 Identify

This verb is used to retrieve information about a repository.

http://baseURL?verb=Identify

Arguments
None

Error and Exception Conditions

badArgument	The request includes illegal arguments.

Response Format

The response must include one instance of the following elements:

repositoryName	a human readable name for the repository
baseURL	the base URL of the repository
protocolVersion	the version of the OAI-PMH supported by the repository: 2.0

earliestDatestamp	a date time that is the guaranteed lower limit of all datestamps recording changes, modifications, or deletions in the repository. A repository must not use date stamps lower than the one specified by the content of the earliestDatestamp element. earliestDatestamp must be expressed at the finest granularity supported by the repository.
deletedRecord	the manner in which the repository supports the notion of deleted records. Legitimate values are no; transient; persistent.
granularity	the finest harvesting granularity supported by the repository. Legitimate values are: YYYY-MM-DD or YYYY-MM-DDThh:mm:ssZ.

The response must include one or more instances of the following element:

adminEmail	the e-mail address of an administrator of the repository

It implies a data provider should have defined first all these parameters identifying a repository:

- **the repository name**
- **the repository base URL**
- **the way deleted records are managed**
- **the granularity**
- **one or several repository administrator email adresses**

XML response description

Below is an example of what could be the response to an Identify request sent to a repository which:

baseURL is	http://www.aMuseum.org/cgi-bin/oai
repositoryName is	repositoryX
adminEmail is	administrator@aMuseum.org
deletedRecord is	transient
granularity is	YYYY-MM-DD
earliestDatestamp is	2009-11-26

Identify request:

http://www.aMuseum.org/cgi-bin/oai/verb=Identify

XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH
        xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
                            http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">

        <responseDate>
                2010-01-20T18:30:00Z
        </responseDate>

        <request verb="Identify">
                http://www.aMuseum.org/cgi-bin/oai
        </request>

        <Identify>
                <repositoryName>repositoryX</repositoryName>
                <baseURL>
                        http://www.aMuseum.org/cgi-bin/oai
                </baseURL>
                <protocolVersion>2.0</protocolVersion>
                <adminEmail>administrator@aMuseum.org</adminEmail>
                <earliestDatestamp>2009-11-26</earliestDatestamp>
                <deletedRecord>transient</deletedRecord>
                <granularity>YYYY-MM-DD</granularity>
        </Identify>
</OAI-PMH>
```

20.2 ListMetadataFormats

This verb is used to retrieve the metadata formats available from a repository.

http://baseURL?verb=ListMetadataFormats[&identifier=*an item identifier*]

Arguments

identifier	an optional argument that specifies the unique identifier of the item for which available metadata formats are being requested. If this argument is omitted, then the response includes all metadata formats supported by this repository.

Error and Exception Conditions

badArgument	The request includes illegal arguments or is missing required arguments.
idDoesNotExist	The value of the identifier argument is unknown or illegal in this repository.

noMetadataFormats There are no metadata formats available for the specified item.

XML response description

Below is an example of what could be the response to an ListMetadataFormats request sent to a repository which supports the Dublin Core (oai_dc) and LIDO (lido) metadata formats.

ListMetadataFormats request:

http://www.aMuseum.org/cgi-bin/oai/verb=ListMetadataFormats

XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH
        xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
                        http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">

        <responseDate>2010-01-08T14:27:19Z</responseDate>
        <request verb="ListMetadataFormats"
        <ListMetadataFormats>
                <metadataFormat>
                        <metadataPrefix>
                                oai_dc
                        </metadataPrefix>
                        <schema>
                                http://www.openarchives.org/OAI/2.0/oai_dc.xsd
                        </schema>
                        <metadataNamespace>
                                http://www.openarchives.org/OAI/2.0/oai_dc/
                        </metadataNamespace>
                </metadataFormat>
                <metadataFormat>
                        <metadataPrefix>
                                lido
                        </metadataPrefix>
                        <schema>
                                http://www.lido-schema.org lido-v1-0.xsd
                        </schema>
                        <metadataNamespace>
                                http://www.lido-schema.org
                        </metadataNamespace>
                </metadataFormat>
        </ListMetadataFormats>
</OAI-PMH>
```

20.3 ListIdentifiers

This verb is an abbreviated form of the ListRecords request, retrieving only headers rather than records.

http://baseURL?verb=ListIdentifiers&metadataPrefix=*a metadata prefix*

Arguments

metadataPrefix	a required argument, which specifies that headers should be returned only if the metadata format matching the supplied metadataPrefix is available.

Error and Exception Conditions

badArgument	The request includes illegal arguments or is missing required arguments.
cannotDisseminateFormat	The value of the metadataPrefix argument is not supported by the repository.
noRecordsMatch	The combination of the values of the from, until, and set arguments results in an empty list.

XML response description

Below is an example of what could be the response to an ListIdentifiers request sent to a repository supporting LIDO (lido) metadata format.
The response contains N record headers. For each record, the header delivers:

- the record identifier
- the record date stamp (granularity is YYYY-MM-DD)

ListIdentifiers request:

http://www.aMuseum.org/cgi-bin/oai/verb=ListIdentifiers&metadataPrefix=lido

XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH
        xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
                            http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">

        <responseDate>2010-06-01T19:20:30Z</responseDate>
        <request verb="ListIdentifiers" metadataPrefix="lido"
        <ListIdentifiers>
                <header>
                        <identifier>record 1 identifier</identifier>
                        <datestamp>2008-06-01</datestamp>
                </header>
```

```
                    ….
                    <header>
                        <identifier>record N identifier</identifier>
                        <datestamp>2009-10-15</datestamp>
                    </header>
                </ListIdentifiers>
            </OAI-PMH>
```

20.4 ListRecords

This verb is used to harvest records from a repository.

http://baseURL?verb=ListRecords&metadataPrefix=*a metadata prefix*

Arguments

metadataPrefix	a required argument that specifies the metadataPrefix of the format that should be included in the metadata part of the returned records.

Error and Exception Conditions

badArgument	The request includes illegal arguments or is missing required arguments.
cannotDisseminateFormat	The value of the metadataPrefix argument is not supported by the repository.
noRecordsMatch	The combination of the values of the from, until, set and metadataPrefix arguments results in an empty list.

XML response description

Below is an example of what could be the response to an ListRecords request sent to a repository supporting LIDO (lido) metadata format.
The response returns N records. For each record, the response delivers:

- the record identifier
- the record date stamp (granularity is YYYY-MM-DD)
- the metadata contained in the record

ListRecords request:

http:// www.aMuseum.org/cgi-bin/oai/verb=ListRecords&metadataPrefix=lido

XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH
        xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
                        http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">

<responseDate>2010-06-01T19:20:30Z</responseDate>
<request verb="ListRecords" metadataPrefix="lido">
        http://www.aMuseum.org/cgi-bin/oai
</request>

<ListRecords>
        <record>
                <header>
                        <identifier>record 1 identifier</identifier>
                        <datestamp>2008-12-25</datestamp>
                </header>
                <metadata>
                        <lido:lido
                                xmlns:lido=
                                        "http://www.openarchives.org/OAI/2.0/lido/"
                                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:schemaLocation=
                                        "http://www.openarchives.org/OAI/2.0/lido/
                                        http://www.openarchives.org/OAI/2.0/lido.xsd
                        >
                                <lido:lidoWrap>
                                        record 1 content
                                </lido:lidoWrap>
                        </lido:lido>
                </metadata>
        </record>
        …

        <record>
                <header>
                        <identifier>record N identifier</identifier>
                        <datestamp>2009-12-17</datestamp>
                </header>
                <metadata>
                        <lido:lido
                                xmlns:lido=
                                        "http://www.openarchives.org/OAI/2.0/lido/"
                                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:schemaLocation=
```

```
                                    "http://www.openarchives.org/OAI/2.0/lido/
                                    http://www.openarchives.org/OAI/2.0/lido.xsd>
                              <lido:lidoWrap>
                                    record N content
                              </lido:lidoWrap>
                        </lido:lido>
                  </metadata>
            </record>
</ListRecords>
</OAI-PMH>
```

20.5 GetRecord

This verb is used to retrieve an individual metadata record from a repository. Required arguments specify the identifier of the item from which the record is requested and the format of the metadata that should be included in the record.

Arguments

identifier	a required argument that specifies the unique identifier of the item in the repository from which the record must be disseminated.
metadataPrefix	a required argument that specifies the metadataPrefix of the format that should be included in the metadata part of the returned record.

Error and Exception Conditions

badArgument	The request includes illegal arguments or is missing required arguments.
cannotDisseminateFormat	The value of the metadataPrefix argument is not supported by the item identified by the value of the identifier argument.
idDoesNotExist	The value of the identifier argument is unknown or illegal in this repository.

XML response description

Below is an example of what could be the response to a GetRecord request sent to a repository supporting LIDO (lido) metadata format.
The requested record is record 1.
The response returns:

- the record identifier
- the record date stamp (granularity is YYYY-MM-DD)
- the metadata contained in the record

GetRecord request:

http://www.aMuseum.org/cgi-bin/oai/verb=GetRecord&metadataPrefix=lido
&identifier=record 1 identifier

XML response:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH
        xmlns="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
                            http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd"
>
<responseDate>2010-02-08T08:55:46Z</responseDate>
<request verb="GetRecord" identifier="record 1 identifier" metadataPrefix="lido">
        http://www.aMuseum.org/cgi-bin/oai
</request>
<GetRecord>
        <record>
                <header>
                        <identifier>record 1 identifier</identifier>
                        <datestamp>2009-08-14</datestamp>
                </header>
                <metadata>
                        <lido:lido
                                xmlns:lido=
                                        "http://www.openarchives.org/OAI/2.0/lido/"
                                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:schemaLocation=
                                        "http://www.openarchives.org/OAI/2.0/lido/
                                        http://www.openarchives.org/OAI/2.0/lido.xsd>
                                <lido:lidoWrap>
                                        record 1 content
                                </lido:lidoWrap>
                        </lido:lido>
                </metadata>
        </record>
</GetRecord>
</OAI-PMH>
```

21 Static Repository XML file example

```
<?xml version="1.0" encoding="UTF-8"?>

<Repository
        xmlns="http://www.openarchives.org/OAI/2.0/static-repository"
        xmlns:oai="http://www.openarchives.org/OAI/2.0/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/static-repository
                            http://www.openarchives.org/OAI/2.0/static-repository.xsd"
>

<Identify>
        <oai:repositoryName>Demo repository</oai:repositoryName>
        <oai:baseURL>http://gateway.WebsiteX.org/www.MuseumA.org/repositoryA.xml
        </oai:baseURL>
        <oai:protocolVersion>2.0</oai:protocolVersion>
        <oai:adminEmail>admin@museumA.org</oai:adminEmail>
        <oai:earliestDatestamp>2008-09-19</oai:earliestDatestamp>
        <oai:deletedRecord>no</oai:deletedRecord>
        <oai:granularity>YYYY-MM-DD</oai:granularity>
</Identify>

<ListMetadataFormats>
        <oai:metadataFormat>
                <oai:metadataPrefix>
                        oai_dc
                </oai:metadataPrefix>
                <oai:schema>
                        http://www.openarchives.org/OAI/2.0/oai_dc.xsd
                </oai:schema>
                <oai:metadataNamespace>
                        http://www.openarchives.org/OAI/2.0/oai_dc/
                </oai:metadataNamespace>
        </oai:metadataFormat>
        <oai:metadataFormat>
                <oai:metadataPrefix>
                        lido
                </oai:metadataPrefix>
                <oai:schema>
                        http://www.lido-schema.org lido-v1-0.xsd
                </oai:schema>
                <oai:metadataNamespace>
                        http://www.lido-schema.org
                </oai:metadataNamespace>
```

```
        </oai:metadataFormat>
</ListMetadataFormats>

<ListRecords>
        <oai:record>
                <oai:header>
                        <oai:identifier>record 1 identifier</oai:identifier>
                        <oai:datestamp>2009-03-02</oai:datestamp>
                </oai:header>
                <oai:metadata>
                        <lido:lido
                                xmlns:lido=
                                        "http://www.openarchives.org/OAI/2.0/lido/"
                                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:schemaLocation=
                                        "http://www.openarchives.org/OAI/2.0/lido/
                                        http://www.openarchives.org/OAI/2.0/lido.xsd
                        >
                                <lido:lidoWrap>
                                        record 1 content
                                </lido:lidoWrap>
                        </lido:lido>
                </oai:metadata>
        </oai:record>
        …
        <oai:record>
                <oai:header>
                        <oai:identifier>record N identifier</oai:identifier>
                        <oai:datestamp>2009-12-29</oai:datestamp>
                </oai:header>
                <oai:metadata>
                        <lido:lido
                                xmlns:lido=
                                        "http://www.openarchives.org/OAI/2.0/lido/"
                                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                xsi:schemaLocation=
                                        "http://www.openarchives.org/OAI/2.0/lido/
                                        http://www.openarchives.org/OAI/2.0/lido.xsd
                        >
                                <lido:lidoWrap>
                                        record N content
                                </lido:lidoWrap>
                        </lido:lido>
                </oai:metadata>
        </oai:record>
</ListRecords>
</Repository>
```

In this case too, a data provider should have defined first all these parameters identifying a static repository:

- **the repository name**
- **the repository base URL (which must be the URL of the XML file describing the static repository combined with the URL of the static repository gateway)**
- **the way deleted records are managed must be set to "no"**
- **the granularity**
- **one or several repository administrator email adresses**

22 OAI-PMH 2.0 specification and implementation documents

A tutorial for beginners
http://www.oaforum.org/tutorial/english/intro.htm

The Open Archives Initiative Protocol for Metadata Harvesting v2.0 specification
<http://www.openarchives.org/OAI/openarchivesprotocol.html>

Implementing OAI-PMH
http://www.openarchives.org/OAI/2.0/guidelines.htm

Implementing OAI-PMH static repositories and static repository gateway
http://www.openarchives.org/OAI/2.0/guidelines-static-repository.htm

23 OAI-PMH 2.0 toolkits

Web sites offering free OAI-PMH implementations

OAICat Open Source Software (OSS)
<http://www.oclc.org/research/software/oai/cat.htm>

ibiblio's PHP, OAI-PMH Data Provider Implementation
<http://www.ibiblio.org/oaibiblio/>

ruby-oai
<http://oai.rubyforge.org/>

OAI PMH Tools (list of tools)
<http://www.openarchives.org/pmh/tools/tools.php>

Rapid Visual OAI Tool
<http://rvot.sourceforge.net>

Example of a Very large OAI implementation

OAIster
<http://www.oaister.org>

The following site presents an interface to interactively test repositories for compliance with the OAI Protocol for Metadata Harvesting

<http://www.purl.org/NET/oai_explorer>