MPEG21 DIDL Application Profile for Institutional Repositories

Table of contents

- Table of contents
- Document information
- Document History
- Abstract
- Introduction
- Compound and Digital Objects as Digital Items
- OAI Response with a DIDL document
- DIDL as a wrapper
    - Item Descriptors
        - Item 'Identifier' Statement
        - Item 'modified' Statement
        - Item 'type' Statement
    - The Top Item as a Compound Object
    - Second level Items
        - Metadata Item
        - Object File Item
        - Jump-off-page / Human Start page Item
- Example of full OAI-PMH record with a MPEG-21 DIDL document

Document information

Title: MPEG21 DIDL Application Profile for Institutional Repositories
Subject: MPEG21 DIDL
Moderator: Thomas Place
Version: 3.0
Date published: 2009-04-18
Excerpt: Write an excerpt

(Optional information)
Type:
Format:
Identifier:
Language:
Rights:
Tags:

Document History

Date	Version history	Owner	Changelog	PDF

18 April 2009	3.0	SURFshare	Start of version 3.0. Based on NEEO document "MPEG21 DIDL Application Profile for Institutional Repositories" version 0.4, which is based on "MPEG21 DIDL Document Specifications for repositories" version 2.3.1. See also history. Note that this is the first version of this document. The version number (3.0) indicates that it is more recent and more up-to-date than the predecessors on which it is based by having a higher number than their latest versions. Changes: 1. Less strict remarks on the location of the namespace declarations. 2. The number of Descriptors is unlimited. 3. Type of a Digital Item is expressed as an URI being the value of the rdf:resource attribute of a rdf:type element. 4. Type statements are also allowed at the top level. 5. Instead of using not yet registered terms from info:eu-repo/semantics/, the terms from the Eprints Access Rights Vocabulary must be used for expressing the accessibility of an object 6. For the date of deposit of an object in a repository **dcterms:dateSubmitted** is to be used (was **dcterms:issued**)	Download
22 Januari 2008	2.3.1	SURFshare	Minor change in the schema path. ISO changed the path .../dii.xsd/dii.xsd to .../dii/dii.xsd Minor change in the examples the mimetype for humanStartPage resources changed from "application/html" to "text/html". "application/html" is not a valid mimetype.	Download
05 December 2007	2.3	SURFshare	Changes to stress the use of Persistent Identifiers in the DIDL document. The addition of the ORE compliant info:eu-repo namespace	
23 May 2007	2.2.2	SURFshare	Some changes and little tweaks.	
23 March 2007	2.2.1	SURFshare	Added comment of Peter van Huisstede, small corrections in the example XML.	
6 March 2007	2.2	SURFshare	The Committee for Complex Objects looked at this document and came with more elegant improvements. Thanks to: Thomas Place, Renze Brandsma, Henk Ellermann, Peter van Huisstede and Ruud Bronmans.	
20 February 2007	2.1	SURFshare	A closer look at the recommendations of Herbert vd Sompel gave more insight in the DIDL semantics, and thus leading to a better XMLspecification.	
2 January 2007	2.0	SURFshare	Fundamental change of element and attribute use; for better representation of the semantics. Additional texts for driver guidelines from Martin Feijen, new DIDL according to comments of Herbert vd Sompel, new DIDL schema. (http://purl.lanl.gov/STB-RL/schemas/2006-09/DIDL.xsd)	
4 December 2006	1.1.2	SURFshare	Translated into English for DRIVER	
11 July 2006	1.1.1	SURFshare	Few typos are removed.	
10 July 2006	1.1	SURFshare	Extension with: • Version numbering and information • Complete namespace declaration in 'metadata'-item. • The three Items are not case sensitively discriminated by: metadata, objects and jump-off-page. • Extended explanation about the use of Namespace-declarations.	
30 March 2006	1.0	SURFshare	Initial document	

	0.4	NEEO	1. Some minor editorial changes 2. 3.3.2 - Object File Item • Addition of deposit date as a <dcterms:issued> • Vocabulary for <dcterms:accessRights>	
	0.3	NEEO	Only minor changes	
	0.2	NEEO	1. One or more object files is changed in zero, one or more object files 2. Removed example; has to be replaced by a new example.	
	0.1	NEEO	Changes with respect to version 2.3.1 of "MPEG21 DIDL Document Specifications for repositories" by Maurice Vanderfeesten (SURF) 1. Relaxed the requirement that the top Item identifier must be persistent. 2. More strict formulation for Item identifiers not being the same as the OAI identifier and the DIDL document identifier. 3. Replaced dip:objectType by rdf:type. 4. The use of simple Dublin Core is not mandatory and is not even recommended. 5. MODS is the recommended metadata scheme. 6. Left out all references to 'work' or 'expression'. 7. Item elements stand for Digital Items (they are Digital Item Declarations). Type statements (using rdf: type) type the Digital Items. 8. The use of identifiers (URIs) for Digital Items is mandatory. 9. Added new semantics: publishedVersion|authorVersion, embargo, description, access rights.	

Abstract

The abstract describes what the application profile is about. It should contain a problem definition, the standards described by the application profile and the goal of the application profile.

Introduction

This document is an adaptation of "MPEG21 DIDL Application Profile for NEEO" version 0.4 (http://drcwww.uvt.nl/~place/neeo/didl%20application%20profile.0.4.doc). The latter was based on Maurice Vanderfeesten (2008), "MPEG21 DIDL Document Specifications for repositories" version 2.3.1 https://www.surfgroepen.nl/sites/oai/complexobjects/Shared%20Documents/DIDLdocumentSpecification_EN_v2.3.doc

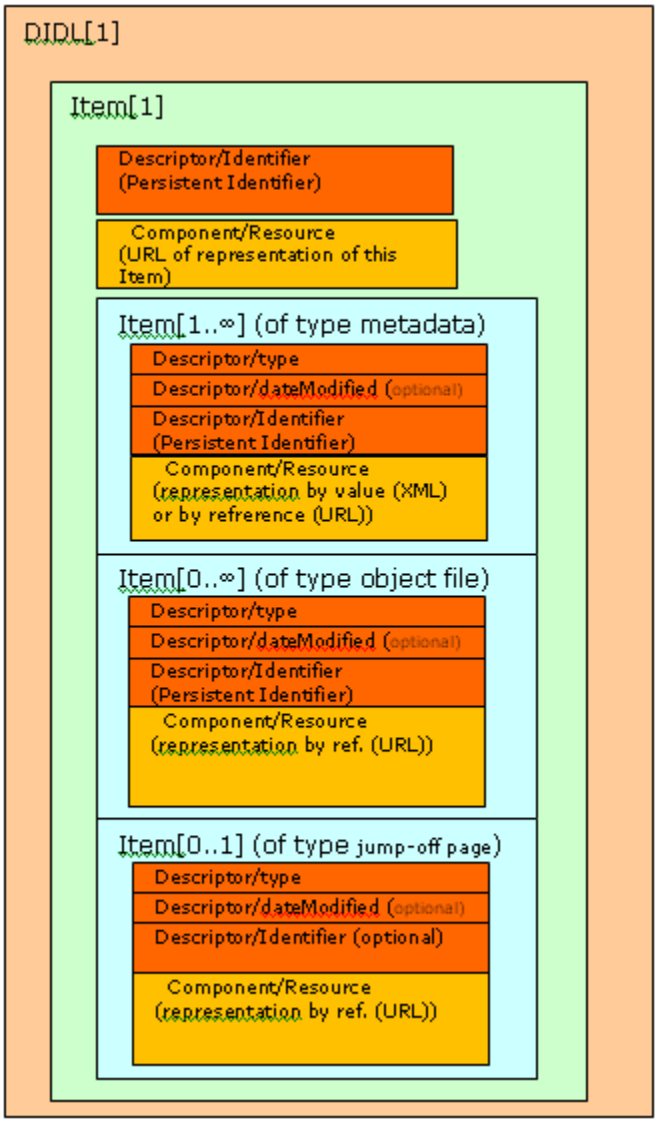
This document describes the use of DIDL in the context of institutional repositories. The DIDL Document Specification was originally developed within the DARE programme of SURF as a solution for:

- the harvesting of the digital resources (PDFs etc.) from the local repositories for ingest into the E-Depot system of the Royal Library for long term preservation
- the harvesting of the digital resources (PDFs etc.) from the local repositories by a service provider (e.g. a search portal that indexes the full text of documents)
- the representation of complex documents such as doctoral theses that consist of multiple digital resource files
- the confusing use of dc:identifier; dc:identifier can be used for different types of objects; the identifier itself doesn't indicate what type of object is identified; in the context of repositories dc:identifier can point to a jump-off page (JOP) and/or to object files.

DIDL has been in use by the DARE community since the summer of 2006. One of the results is that the content of all Dutch repositories are now part of the E-Depot of the Royal Library, the national library of The Netherlands.

Compound and Digital Objects as Digital Items

The digital objects that populate institutional repositories can be seen as compound objects that consist of parts or components that are also digital objects. In the DIDL model the basic entity is a Digital Item. The compound objects and their objects play the role of Digital Items in the model that underlies DIDL. In a DIDL document the Item elements represent the Digital Items. The top Item element that is situated directly below the DIDL root element is used for the compound object. The Item elements that are the children of the top Item element represent the objects that are part of the compound object. The objects that are part of a compound object can themselves be a compound object. When a part object is also a compound object, then its parts are not described in the same DIDL document, but a separate DIDL document is used to describe this compound object with its parts. This means that in this application profile there are only two levels of Digital Items within a DIDL document. Although DIDL allows for a hierarchy of Digital Items, this profile restricts the hierarchy to two levels: the level of the top Digital Item, the compound object and the level of the Digital Items that are parts of the top Digital Item. This version of the application profile doesn't give (yet) guidelines for the case of a compound object that is part of another compound object.
This profile distinguishes three types of digital objects: descriptive metadata, object files and jump-off pages. This list is extensible; other types can be added.
The figure below is a schematic representation of a DIDL document of a compound object that consists of one or more descriptive metadata records, zero or more object files and zero or one jump-off page. Metadata that apply to the metadata records, object files and jump-off pages can be placed in Descriptor elements within the respective Item elements. In the figure the most used Descriptors are shown. The list of Descriptor elements in an Item is extensible.
A digital object can have one or more representations. A representation is the thing that can be displayed on a computer screen or that can be printed. A representation MUST have a medium type (mimetype). In DIDL, representations are handled by the Resource element. A Resource is contained in a Component element which in its turn is a child of the Item element. There are two ways of including a representation in a Resource element. The first way is **by-value**: the representation as such is included as content of the Resource element. This is the usual way that metadata records formatted in XML are included. The second way is **by-ref**: the Resource element stays empty, but the representation is referred to by an URL that is the value of the ref-attribute of the Resource element. Normally, the URL will point to a file in the repository.
Each Digital Item MUST have an identifier with the exception of jump-off pages for which the identifier is optional. This identifier MUST be an URI. The URI of a Digital Item should be different from the URLs of its representations. The identifiers of the Digital Items must be persistent. The URLs of the representations and the medium types can change, while the identifier of the Digital Item stays the same. This allows, e.g., for replacing a file that can only be processed by an old-fashioned word processor by a version with the same content that can be read at all contemporary desk tops. Or a file can be moved to another location; the identifier of the Digital Items stays the same indicating that it is still the same file. If the policy of a repository is to preserve the different representations of a Digital Item then the repository is advised to treat the representations as separate Digital Items, each with its own persistent identifier. So it is possible that in one repository the PDF, the Word and the HTML versions of a publication are combined into one Digital Item, while in another repository they are treated as separate Digital Items. Another use of Digital Item identifiers is to relate Digital Items to each other.



OAI Response with a DIDL document

The DIDL document is part of an OAI-PMH response. The DIDL document will be returned within an OAI-record when using **didl** as value of the metadataPrefix verb. This enables the repository to generate this particular didl format that is described in the document below.
Within the OAI XML structure, the DIDL resides within the metadata element. See below:

```
<OAI-PMH ...>
...
<request ... metadataPrefix="didl">
...
<record>
<header>...</header>
<metadata>
<didl:DIDL
xmlns:didl="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="
urn:mpeg:mpeg21:2002:02-DIDL-NS
[http://standards.iso.org/ittf/PubliclyAvailableStandards/]
MPEG-21_schema_files/did/didl.xsd

urn:mpeg:mpeg21:2002:01-DII-NS
[http://standards.iso.org/ittf/PubliclyAvailableStandards/]
MPEG-21_schema_files/dii/dii.xsd">
...
</didl:DIDL>
</metadata>
<about>...</about>
</record>
...
</OAI-PMH>
```

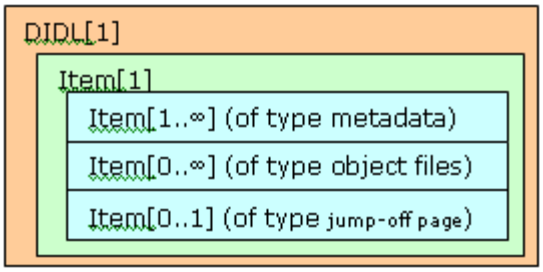
Remarks:

1. Don't forget the **DIDL** tag in the OAI-PMH response.
    a. Make a declaration of the **didl**, **dii**, **dc** and **dcterms** namespaces here, in the DIDL tag.
2. The **about** element is optional in OAI-PMH

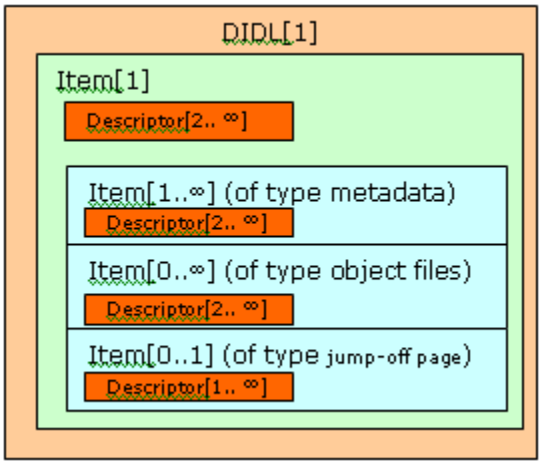
DIDL as a wrapper

The DIDL document is a document with one top-level **Item** element. The **Item** contains several child **Item** elements. These child **Item** elements describe three different types: descriptive metadata, object files and jump-off pages. Between brackets the cardinality of the XML elements are shown:

```
<metadata> \\
<didl:DIDL ...> \\
<didl:Item> \\
<didl:Item>...</didl:Item> \\
<didl:Item>...</didl:Item> \\
<didl:Item>...</didl:Item> \\
</didl:Item> \\
</didl:DIDL> \\
</metadata> \\ \\
\\
```



Item Descriptors



Item Descriptors provide information about the Digital Item. A Descriptor contains a Statement with information about the Item. For each "statement" a new Descriptor is used.
The top level **Item** element **MUST** contain *two* **Descriptor** elements. One **Descriptor** element for the (Persistent) Identifier and one **Descriptor** element for the modification date.

1. Modifications **MUST** be made visible by changing the modification date. When there are no modifications the modification date can be left out from the second level Items.
2. Changes of the modification date in child **Item** elements **MUST** be propagated to the parent **Item** element.
3. When a Descriptor element for modification date is used **also** a **Descriptor** element with an identifier **MUST** be used (they go in pairs). Rationale: In order to compare similar harvested **Item** elements wrt modification date, an identifier must be added.
4. For the second level **Item** elements:
    a. the "type" **Descriptor** element **MUST** always be used
    b. the "identifier" **Descriptor** element **MUST** be used in the metadata and objectfile **Descriptor** elements. This is optional for the jump-off page **Descriptor** element
    c. the "modification date" **Descriptor** element **MAY** be used in all of the second level **Item** elements.

Example on level one	<didl:DIDL ...> \\ <didl:Item> \\ <didl:Descriptor>...</didl:Descriptor> <\!\--Identification, mandatory-\-> \\ <didl:Descriptor>...</didl:Descriptor> <\!\--Modification date, mandatory-\-> \\ <didl:Item>...</didl:Item> \\ <didl:Item>...</didl:Item> \\ <didl:Item>...</didl:Item> \\ ... \\ </didl:Item> \\ </didl:DIDL> \\ \\ \\
Example on level two Object type added	<didl:DIDL ...> \\ <didl:Item> \\ <didl:Item> \\ <didl:Descriptor>...</didl:Descriptor> <\!\--Identification, mandatory-\-> \\ <didl:Descriptor>...</didl:Descriptor> <\!\--Modification date, optional-\-> \\ <didl:Descriptor>...</didl:Descriptor> <\!--\--Type, mandatory-\-> \\ ... \\ </didl:Item> \\ <didl:Item>...</didl:Item> \\ <didl:Item>...</didl:Item> \\ <didl:Item>...</didl:Item> \\ ... \\ </didl:Item> \\ </didl:DIDL> \\ \\ \\

Apart from the Identifier, modified date and type, Descriptors with other semantic content can be used, see section 3.2.

Item 'Identifier' Statement

The *first* **Descriptor** contains the ID of the **Item** elements. This is used to uniquely identify the digital object (e.g. with an urn:nbn). This ID is wrapped in a **Statement** with a **DII Identifier** element. For example:

```
 <didl:Item>
<didl:Item>
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<dii:Identifier>urn:nbn:nl:ui:13-6748398729821</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
...
</didl:Item>
...
</didl:Item>
```

Remarks:

1. In this example the root Item has a Persistent Identifier.
2. The identifier MUST be an URI. Some repositories can make use of registered URIs, e.g., in the Netherlands the repositories can make use of National Bibliography Numbers. If a repository doesn't have access to registered URIs, UUIDs or TAGs can be used. More information about UUIDs can be found at [http://en.wikipedia.org/wiki/UUID|http://en.wikipedia.org/wiki/UUID] and for TAGs the best starting point is http://www.taguri.org/. TAGs for the repository of Tilburg University could be constructed in the following way. All TAGs start with 'tag:uvt.nl,<year>:<record id>'. <year> is the year that the compound object entered the repository. The URIs for MODS records can look like 'tag:uvt.nl,<year>:<record id>/mods' and the URIs for object files can look like 'tag:uvt.nl,<year>:<record id>/<file id>'.
3. The identifiers of the root Item and the second level **Item** elements SHOULD NOT be equal to the OAI identifier or the DIDL document identifier.
4. Identifiers should not change. Different identifier implies a different object.
5. The namespace for **dii** SHOULD be declared in the DIDL tag.

Item 'modified' Statement

The second **Descriptor** contains a modification date. When something changes inside an **Item**, this modification date element has to be updated. Modification date is mandatory in the top level Item and is optional in the second level Items. This modification date is being specified by the **modified** element from **dcterms**:

```
 <didl:Item>
<didl:Item>
...
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<dcterms:modified>2006-12-20T10:29:12Z</dcterms:modified>
</didl:Statement>
</didl:Descriptor>
...
</didl:Item>
...
</didl:Item>
```

Remarks:

1. Declare the **dcterms** namespace in the DIDL tag.
2. The format of the date is Zulu-time; which means that it can be sorted as text.
3. There can be only one **Statement** element in a **Descriptor** element, which means that **dii:identifier** and **dcterms:modified** reside in separate **Descriptor** elements.

Item 'type' Statement

(In Maurice Vanderfeesten (2008), "MPEG21 DIDL Document Specification for repositories", version 2.3.1, the dip:ObjectType is used. Here, this is replaced by rdf:type as more appropriate. For compatibility with Driver and SURFshare both Descriptors can be used. In "MPEG21 DIDL Application Profile for NEEO Repositories" the URI is placed as a literal in the content of the rdf:type element. This is not in line with the use of rdf. Service providers should be aware of these different versions of expressing the type of a Digital Item.)

The third **descriptor** contains the Digital Item type. This type is mainly used the second level Item elements, however, it is also possible to type the top Digital Item.

```
 <didl:Item>
<didl:Item>
...
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<rdf:type rdf:resource="info:eu-repo/semantics/descriptiveMetadata" />
</didl:Statement>
</didl:Descriptor>
...
</didl:Item>
...
</didl:Item>
```

See for more information about the type statement the next section.

Remarks:

1. Declare the **rdf** namespace in the DIDL tag.
2. The value of the **rdf:resource** attribute of the **rdf:type** element is an URI.
3. For typing the Digital Items the info:eu-repo namespace is used.

The Top Item as a Compound Object

The top-**Item** element contains *one mandatory* **Item** sub-element that describes a Digital Item of type 'info:eu-repo/semantics /descriptiveMetadata'. There can be more Digital Items that are descriptive metadata or that are object files.
Optionally there can be a **Item** sub-element that describes a Digital Item of a third type: 'info:eu-repo/semantics/humanStartPage'. A Digital Item of this type is a jump-off-page, i.e., an html intermediate page that describes in a human readable way which objects are involved. In this way a reader can be informed about the fact that a file is available in different formats such as PDF, MS Word or HTML, or that a dissertation consists of separate files (e.g. when the thesis consists of a set of previously published articles).

```
 <didl:DIDL ...>
<didl:Item>
<didl:Item>...</didl:Item> <\!-\- metadata -->
<didl:Item>...</didl:Item> <\!-\- object files -->
<didl:Item>...</didl:Item> <\!-\- jump-off-page -->
</didl:Item>
</didl:DIDL>
```

The DIDL document contains at least one metadata **Item** element. This metadata can be in different formats, simple Dublin Core, qualified Dublin Core, MODS, MARC21, etc. The metadata can be included by-value or can be pointed to by-reference. one of the metadata **Item** elements MUST contain MODS, and the MODS record MUST be included by-value.

```
 <didl:Item>

<didl:Item> <\!-\--one or many occurrences-\->
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<rdf:type rdf:resource="info:eu-repo/semantics/descriptiveMetadata" />
</didl:Statement>
</didl:Descriptor>
...
</didl:Item>

<didl:Item>    <\!-\--zero or many occurrences-\->
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<rdf:type rdf:resource="info:eu-repo/semantics/objectFile" />
</didl:Statement>
</didl:Descriptor>
...
</didl:Item>

<didl:Item> <\!-\--zero or one occurrences-\->
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<rdf:type rdf:resource="info:eu-repo/semantics/humanStartPage" />
</didl:Statement>
</didl:Descriptor>
...
</didl:Item>

</didl:Item>
```

The URIs will be processed case un-sensitive. It is recommended to use camelCase writing. It is VERY important to use the exact combinations of characters, otherwise automatic processing will not be possible. To make it very clear the following URIs are used:

1. info:eu-repo/semantics/descriptiveMetadata (This Item occurs 1 or many times)
2. info:eu-repo/semantics/objectFile (This Item occurs 0 or many times)
3. info:eu-repo/semantics/humanStartPage (This Item occurs 0 or 1 time)

Remarks:

1. The info:eu-repo namespace is used with the following syntax:info:eu-repo/*type*/*identifier* like in info:lanl-repo. For more information see http://info-uri.info/registry/OAIHandler?verb=GetRecord&metadataPrefix=reg&identifier=info:lanl-repo/
2. The semantics of the type statements is to indicate the type of the Digital Item.

Second level Items

Metadata Item

When the metadata are included by-value in an **Item** element, then the metadata form the content of a **Resource** element. The case of by-reference is described in the "Object File Item" section. The **Resource** element is contained by a **Component** element. If there are several representations of the same metadata record, e.g., a version in MODS and a version in MARCXML, it is recommended to use separate **Item** elements for each representation.
MODS is mandatory; the MODS records MUST be included by-value. Notice that the guidelines of Driver still mention Simple Dublin Core. To be compliant with both the present Application Profile and Driver, separate metadata Items must be included, one for MODS and the other for Dublin Core.

```
  <didl:Item>
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<rdf:type rdf:resource="info:eu-repo/semantics/descriptiveMetadata" />
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\- This metadata instance has its own ID number -->
<didl:Statement mimeType="application/xml">
<dii:Identifier>urn:nbn:nl:ui:13-74836724783</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\- This record has its own Modification date -->
<didl:Statement mimeType="application/xml">
<dcterms:modified>2006-12-20T10:29:12Z</dcterms:modified>
</didl:Statement>
</didl:Descriptor>
<didl:Component>
<didl:Resource mimeType="application/xml"> <\!-\- the MODS data -->
<mods:mods
xmlns:mods="http://www.loc.gov/mods/v3"
xsi:schemaLocation=
"http://www.loc.gov/mods/v3
[http://www.loc.gov/standards/mods/v3/mods-3-3.xsd]">
<mods:titleInfo>...</mods:titleInfo>
<mods:name>...</mods:name>
<mods:typeOfResource> ... </mods:typeOfResource>
...
</mods:mods>
</didl:Resource>
</didl:Component>
</didl:Item>
```

Remarks:

1. If the date of the metadata has been changed, make sure that the date of the root-level Item modification date is also being changed.
2. Declare the **mods** namespace in the start-tag of MODS record, i.e., in the **mods** element.

Object File Item

An Object File **Item** contains a link to a digital object. This is 'by_ref', and the **Item** element has a type statement with an **info:eu-repo/semantics/objectFile** URI. An Object File **Item** can occur zero, one or more times.
When there are more representations of the same object file, then this can be handled in two ways:

1. for each representation there is a separate **Resource** element within the same **Component** element. The Descriptors of the **Item** element apply to both representations. The representations only differ in medium type (mimetype).
2. for each representation there is a separate **Item** element. The representations can differ not only in medium type but also in other respects as reflected in their respective Descriptors.

Additional Descriptor elements can be used to describe certain aspects of the object file:

- To indicate whether the file is a (exact copy of the) published version or the version of the author that is accepted by the publisher for publication a Descriptor with a type statement can be used. The proposal is to use the following URIs:
    - info:eu-repo/semantics/publishedVersion
    - info:eu-repo/semantics/authorVersion
      These proposed URIs are not yet officially registered.

- To add descriptions like "Introduction", "Chapter1" and "Glossary", the **dc:description** element can be used within the **Statement** element of a Descriptor. The **dc:description** element can also be used for other (unstructured) information for which there is no specific element.

- The date at which the object file is deposited can be expressed as a **dcterms:dateSubmitted** element.

- If there is an embargo on a file, the **dcterms:available** element can be used with the date that the file will become available.

- Information about the accessibility of an object file should go in the **dcterms:accessRights** element. The vocabulary Following SWAP: http://www.ukoln.ac.uk/repositories/digirep/index/Eprints_AccessRights_Vocabulary_Encoding_Scheme to be used for its content is as follows :
    - http://purl.org/eprint/accessRights/ClosedAccess
    - http://purl.org/eprint/accessRights/RestrictedAccess
    - http://purl.org/eprint/accessRights/OpenAccess

```
<didl:Item>
...

<\!-\- Below this line one can find links to one or more digital objects -->
<didl:Item> <\!-\- First Item for a File/Bitstream -->
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<rdf:type rdf:resource="info:eu-repo/semantics/objectFile" />
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\- This Object Item has its own persistent ID -->
<didl:Statement mimeType="application/xml">
<dii:Identifier>urn:nbn:nl:ui:13-36724783</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\- This Item has its own Modification date -->
<didl:Statement mimeType="application/xml">
<dcterms:modified>2006-12-20T10:29:12Z</dcterms:modified>
</didl:Statement>
</didl:Descriptor>

<didl:Component>
<didl:Resource
mimeType="application/pdf"
ref="http://my.server.nl/report.pdf"/>
</didl:Component>
</didl:Item>

<didl:Item> <\!-\- Second Item for a File/Bitstream -->
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<rdf:type rdf:resource="info:eu-repo/semantics/objectFile" />
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\- This Object Item has its own persistent ID -->
<didl:Statement mimeType="application/xml">
<dii:Identifier>urn:nbn:nl:ui:13-36724784</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\- This Item has its own Modification date -->
<didl:Statement mimeType="application/xml">
<dcterms:modified>2006-12-20T10:29:12Z</dcterms:modified>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\- this file is the appendix -->
<didl:Statement mimeType="application/xml">
<dc:description>Appendix</dc:description>
</didl:Statement>
</didl:Descriptor>
...
<didl:Component>
<didl:Resource
mimeType="application/pdf"
ref="http://my.server.nl/appendix.pdf"/>
</didl:Component>
</didl:Item>

<didl:Item> <\!-\- Third Item for a File/Bitstream -->
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<rdf:type rdf:resource="info:eu-repo/semantics/objectFile" />
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\- This Object Item has its own persistent ID -->
<didl:Statement mimeType="application/xml">
<dii:Identifier>urn:nbn:nl:ui:13-36724785</dii:Identifier>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\- This Item has its own Modification date -->
<didl:Statement mimeType="application/xml">
<dcterms:modified>2006-12-20T10:29:12Z</dcterms:modified>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\--\- deposit date -->
```

```
<didl:Statement mimeType="application/xml">
<dcterms:issued>2010-12-01</dcterms:issued>
</didl:Statement>
</didl:Descriptor>
<didl:Descriptor> <\!-\--\- embargo on file -->
<didl:Statement mimeType="application/xml">
<dcterms:available>2010-12-01</dcterms:available>
</didl:Statement>
</didl:Descriptor>
...
<didl:Component>
<didl:Resource
mimeType="application/pdf"
ref="http://my.server.nl/datasheets.xls"/>
</didl:Component>
</didl:Item>
</didl:Item>
```

In the above example, the Resource locations are not repeated within **Component** element, but each Resource location is wrapped in its own Item element. The rationale behind this is that each Bitstream or file can have its own Identifier and its own Descriptors.

Remarks:

1. The order of the Object File Items should be in logical reading order! The Item with chapter 1 should be followed by the next sibling Item element that contains chapter 2, etc... This way the service provider can make a better presentation. Making the order explicit by placing sequence numbers is being specified in a next version of the specification.
2. If there are important modifications in the **Resource** element or Descriptors of the Item, the modification date must be propagated to the modification date of the root level Item.
3. Use for a **modified** or **Identifier** element a separate <**Descriptor>** <**Statement>** element construction.
4. The rule of thumb is that if a Bitstream or file has its own identifier, the wrapper is an Item element. To keep the possibility open for a Bitstream to have an Identifier, we use the Item element as default to wrap a resource location.
5. For representing dates ISO 8601 MUST be used and more in particular the formats as defined in http://www.w3.org/TR/NOTE-datetime.

Jump-off-page / Human Start page Item

The third ObjectType **Item** element contains a link to the jump-off page or intermediate page. This is done in the same way as for the Object **Item** element. This **Item** element is *optional*. There should not be more than one Item of this type. The **identifier** element and **modified** elements are optional.

```
<didl:Item>
...

<\!-\- Below this line; an Item with a link to one optional Intermediate page -->
<didl:Item>
<didl:Descriptor>
<didl:Statement mimeType="application/xml">
<rdf:type rdf:resource="info:eu-repo/semantics/humanStartPage" />
</didl:Statement>
</didl:Descriptor>
...
<didl:Component>
<didl:Resource
mimeType="text/html"
ref="http://my.server.nl/mypub.html"/>
</didl:Component>
</didl:Item>
</didl:Item>
```

Remarks:

1. Using a Persistent Identifier for this HTML page means that you always need to resolve this page in the distant future. This is not recommended!

Example of full OAI-PMH record with a MPEG-21 DIDL document

⚠ ===== TO DO =====