kopal
Daten für die Zukunft

Project *kopal*
Co-operative Development of a Long-Term Digital Information Archiv

Universal Object Format

An archiving and exchange format for digital objects

By

Dipl.-Inform. Tobias Steinke

Frankfurt am Main 2006

GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

1. Introduction

The present text defines a format with which digital objects along with metadata can be archived and exchanged between institutions and archiving systems. It is based on the formats *Metadata Encoding and Transmission Standard* (METS, Version 1.4) and the *Long-term preservation Metadata for Electronic Resources* (LMER, Version 1.2), and has been developed within the project *kopal*[1]. The archiving system used in kopal is *DIAS*[2]. The version of this software that is being used in the project, conditions certain technical agreements that influence the following descriptions. Such system dependencies will be marked with "***DIAS***" and set in italics.

The description presupposes the knowledge of the formats METS and LMER. That applies especially for the following documents:

- METS: An Overview & Tutorial[3]
- LMER 1.2 – Reference Description[4]

Exemplary files on the Universal Object Format (UOF) can be downloaded from the kopal project website. The free software *kopal Library for Retrieval and Ingest* (koLibRI) with which archive objects can be created according to the UOF, is also available there.

[1] http://kopal.langzeitarchivierung.de/
[2] http://www.ibm.com/nl/dias/
[3] http://www.loc.gov/standards/mets/METSOverview.v2.html
[4] http://nbn-resolving.de/?urn=urn:nbn:de:1111-2005051906

2. Object format

For the UOF, an archive object consists of a packed file that contains an arbitrary folder structure with an arbitrary number of files. At the root level of the folder structure, there must be a file named "mets.xml" representing a valid XML file, according to the METS 1.4 schema[5]. Any paths within the packed file have to be stated relative. There is no special default for the file name of the archive object; it should, however, conform to the requirements of the used file systems and ensure that different archive objects bear different names.

DIAS: *Allowed pack formats are ZIP (compatible to PKZIP version 2.50 and higher but below 5.0; standard compression or uncompressed only), GNU-TAR (up to version 1.15.1) and TAR. One archive object must not contain more than 5.000 files. When using the ZIP format, the embodied files must not exceed the maximum of 2 GB for each file.*

[5] http://www.loc.gov/standards/mets/version14/mets.xsd

3. Structure of the file mets.xml

In principle, valid METS files complying to the following criteria are allowed. Beyond those mentioned in the following paragraphs, an arbitrary number of further metadata is possible; their practical use, however, is highly dependent on the used archiving system.

***DIAS**: In DIAS, the METS file is being archived and delivered together with the other files of the archive object. Only a part of the metadata, however, is being analyzed and checked during the ingest. That part is being stored in the so-called Data Management and can be queried systematically for the purposes of long-term preservation.*

The specifications of METS 1.4 state a number of sections and sub-sections for each METS file. Of those sections only METS Header, Descriptive Metadata (dmdSec), Administrative Metadata (amdSec) with the sub-sections Technical Metadata (techMD) and Digital Provenance Metadata (digiprovMD), as well as File Section and Structural Map are defined for the UOF. The other sections may be used nevertheless as long as they are valid METS 1.4.

For the sections in dmdSec and amdSec, METS 1.4 allows arbitrary metadata if they have a valid XML schema. These metadata may be referenced in external files or directly embedded within the same file. The UOF only allows the latter variant, i.e., all metadata of the archive object must be in the one and only mets.xml file.

In the amdSec sections of METS in the UOF, LMER 1.2 is being used. That exploits the modular approach of LMER 1.2. Only the following LMER XML schemas are being used: lmer-object.xsd, lmer-file.xsd and lmer-process.xsd. That means that in each case only the particular elements and their attributes are being integrated within METS but not the LMER structure as it has been defined in lmer.xsd. It also means that the mandatory fields, as they are listed in the LMER Reference Description, do not apply. In fact, especially those LMER elements are being omitted that already have been described by identical METS elements or METS attributes. Elements from LMER-Object and LMER-File can be found in the METS sections of the category amdSec/techMD, and elements from LMER-Process in the METS sections of the category amdSec/digiprovMD.

The following paragraphs refer to an external ID and an internal ID to designate the archive object. The external ID's are Persistent Identifiers[6] (***DIAS:** URN*), which hence are unequivocal worldwide. Internal ID's are only unequivocal within the same archiving system. Archive objects that are migrations of other archive objects, have an identical external ID but different internal ID's.

[6] http://www.persistent-identifier.de/

3.1. METS Header

Within the <mets> tag in the attribute OBJID is the internal ID if it is an export object from an archive (according to OAIS[7] terminology, a DIP).

DIAS: *The attribute OBJID must be empty (OBJID = "") if it is an import object (according to OAIS, a SIP).*

The attribute CREATEDATE of <metsHdr> must contain a valid date that states the creation or last update of the present metadata in the file mets.xml.

The element <agent> must be existent with the attributes ROLE and TYPE as well as the sub-element <name>. That provides information about the creating institution (in case of a SIP) or the archiving system (in case of a DIP). In the latter case, the internal ID in OBJID can be interpreted within the context.

3.2. Descriptive Metadata (dmdSec)

Within this section metadata can be stored that describe the content of the archive object (e.g., the library catalogue entry). Since long-term archives above all need technical information to perform their role, and as the systems in most cases are connected to optimized catalogues anyway, this section in the UOF is optional. It can occur repeatedly, however (***DIAS****: a maximum of 5*), and thus include metadata with different XML schemas (e.g., DC Simple, MODS, MABxml, etc.).

DIAS: *If there exist dmdSec sections with the MDTYPE = "DC", elements of the Dublin Core Metadata Element Set, Version 1.1 (DC Simple)[8] are being expected. DIAS stores them within the Data Management, thusly they are available for targeted requests.*

Each dmdSec has an ID attribute which must be referenced in the Structural Map of the type "ASSET" (see 3.5).

3.3. Administrative Metadata (amdSec)

In the UOF, one or several amdSec sections are allowed. They include at least one techMD section for metadata on the whole archive object and one techMD section for each file belonging to the object. Furthermore, there can be digiprovMD sections for the whole object and for each file. All techMD and digiprovMD sections must have the ID attribute so that they can be referenced in the File section.

DIAS: *A maximum of 5000 amdSec sections, 5001 techMD sections and 5001 digiprovMD sections is allowed.*

---

[7] Reference Model for an Open Archival Information System: http://www.ccsds.org/documents/650x0b1.pdf

[8] http://www.dublincore.org/documents/dces/

The techMD section for the whole archive object includes elements from LMER-Object. Mandatory is only <persistentIdentifier> that names the external ID. <groupIdentifier> can be used repeatedly. <objectVersion> defines the state as original or migration and should be „1“ for an original. If <startFile> exists, that element contains the value of the ID attribute of the corresponding <file> of the File Section of METS. If existent, <numberOfFiles> must correspond to the number of files listed in the File Section of METS.

The respective techMD section of each file includes elements from LMER-File. Only <format> with an appropriate attribute REGISTRYNAME to identify the used namespace (***DIAS**: URN like urn:diasid:fty:kopal:020050705000000000001 for PDF 1.4 and REGISTRYNAME="DIAS" are being used*) is mandatory. <linkedTo> is repeatable and, like in <startFile>, names the corresponding <file> elements of METS. The following elements from LMER-File should be left out, because they already appear mandatory in the File Section of METS: fileIdentifier, path, name, size, fileDateTime, fileChecksum and mimeType.

In the LMER element <xmlData>, further technical metadata in an arbitrary XML schema (***DIAS**: The corresponding XML schema must be stored locally for the archiving system. In practise, in that place mainly the XML output of the tool JHOVE[9] is being inserted*) can be stated for each file. The element is repeatable.

DigiprovMD sections can exist for the whole object and for certain files. The elements <oldMetadataRecordCreator>, <oldObjectIdentifier> and <oldObjectVersion> can only exist when referring to the object. <oldObjectIdentifier> states the internal ID of the predecessor of the described migration. <oldMetadataRecordCreator> provides the context within which the internal ID is valid.

3.4. File Section

The File Section of METS describes all files that belong to the object via relative links. It contains exactly one <fileGrp> element within which for each file a <file> element exists that in turn has one <Flocat> element each. This results in references named by xlink:href that have to be LOCTYPE="URL“ and have to start with „file://./“ since they are relative links.
The following attributes of <file> are mandatory:

- ID
- MIMETYPE
- CREATED
- SIZE
- CHECKSUM
- CHECKSUMTYPE (***DIAS**: For CHECKSUMTYPE, „SHA-1“ and „MD5“ are allowed only*).

[9] http://hul.harvard.edu/jhove/

In the attribute ADMID of <fileGrp>, the ID of the techMD section that contains metadata on the whole object must be referenced. Additionally, the ID's of the digiprovMD sections that describe migrations of the whole object are stated there. Accordingly, in the attribute ADMID of <file> the ID of the corresponding techMD section with the metadata on the certain file and the ID's of file related digiprovMD sections have to be stated. As for the order of each listing of the ID's in ADMID, the following regulation applies: At first, the ID's of digiprovMD sections are being listed in descending order, starting with the last migration; in the end, the ID of the techMD section (for each <file> or in <fileGrp>, exactly one techMD section can be assigned to only).

3.5. Structural Map

There can be an arbitrary number of Structural Maps. However, there must exist exactly one Structural Map with the attribute TYPE="ASSET" that has the following composition: Only one element with the attribute TYPE="ASSET" exists, and this encloses a list of <fptr> elements for each file of the object. In the respective attribute FILEID, the corresponding ID from the File Section is being referenced. In the attribute DMDID of the element, the ID's of all existing dmdSec sections have to be stated.

Besides the mandatory <fptr> elements, ASSET also can list <mptr> elements relating to external ID's of other archive objects that are linked technically to the present object (e.g., an XML schema that is being referenced by XML files of the present object). This mechanism should not cover content related correlations (e.g., journals); metadata in the dmdSec sections are more suitable for that purpose.