

METS Profiles

- Brian Tingle
- co-chair, METS Editorial Board
- Technical Lead, Digital Special Collections
California Digital Library
University of California Office of the President
- DLF Fall Forum
- Philadelphia, Pennsylvania
- 2007-Nov-08

Outline

- What is a METS Profile, and why are they needed?
- How does one get started with profiles?
- What are the components of a profile? (with examples from registered profiles)
- Profile complaints

What is a METS Profile?

- “METS Profiles are intended to describe a class of METS documents in sufficient detail to provide both document authors and programmers the guidance they require to create and process METS documents conforming with a particular profile.”
- A METS Profile is expressed as an XML document. There is a schema for this purpose.
- Note: A METS Profile is a human-readable prose document and is not intended to be “machine actionable” (at this time).

Why are they needed?

- Profile (engineering) From Wikipedia, the free encyclopedia
- In standardization, a **profile** consists of an agreed-upon subse and interpretation of a specification. Many complex technical specifications have many optional features, such that two conforming implementations may not inter-operate due to choosing different sets of optional features to support. Even when no formal optional features exist within a standard, vendors will often fail to implement (or fail to implement correctly) functionality from the standard which they view as unimportant. ... Also, some writers of standards sometimes produce vague or ambiguous specifications, often unintentionally, but sometimes by intention. The use of profiles can enforce one possible interpretation.
- Users can utilize profiles to ensure interoperability, and in procurement.

Profiles as contracts

- In as much as a profile is successful in describing the METS objects in sufficient detail for developers of METS creation tools and systems that use METS, they can be thought of as contracts between producers and consumers.

What is a “class of documents?”

- Generally two approaches have been pursued
 - Profiles based on content type
 - image or image+text; image; paged text; audio CD; recorded event; web site capture
 - Specific system / specific purpose
 - greenstone; trove; universal object format (kopal); simple object; general object; generic preservation and repository object (ECHO Dep); 7train

How does one get started with Profiles?

- de facto profiles
- unregistered profiles
- <http://www.loc.gov/standards/mets/mets-profiles.html> This page contains the current documentation on profiles and the XML Schema for profiles
- Most people seem to start with the METS documents first and then write a profile that describes the METS

What are the components of a profile?

- Information about the profile
- External/Extension schema
- Description Rules
- Controlled vocabularies
- Structural requirements
- Technical requirements
- Tools
- At least one sample document



Element <METS_Profile>

- Root element for a METS Profile
- may contain | <URI> | <title> | <abstract> | <date> | <contact> | <registration_info> | <related_profile>
- <extension_schema> | <description_rules> | <controlled_vocabularies> | <structural_requirements> | <technical_requirements> | <tool> | <Appendix>

Information about the profile

- Unique URI (assigned by METS Board / LOC)
 - PROFILE="this URI" in root mets element
- Short Title
- Abstract
- Date and time of creation
- Registration information (filled out by Board/loc)
- Contact Information (for profile author)
- Related profiles

Element <extension_schema>

- A profile which will be registered with the Network Development and MARC Standards Office must identify all extension schema which may be used in constructing a METS document conformant with the profile. Extension schema for registered profiles MUST be publicly available.
- The schema must be identified in sufficient detail to allow a document author previously unfamiliar with the schema to unambiguously identify and retrieve it.

Element <extension_schema>

- Those registering profiles with the Network Development and MARC Standards Office are strongly encouraged to include a URI for each identified extension schema which may be used to retrieve that schema from any Internet workstation and to include the full text of all identified extension schema as an appendix to their profile registration.
- Registered schema should also use the context subelement to name the elements within a conforming METS instance where the extension schema may be used. If a profile does not dictate the use of any extension schema, it should contain a single <extension_schema> element with subelement of <note> stating that no extension schema are specified in this profile.
- may contain | [<name>](#) | [<URI>](#) | [<context>](#) | [<note>](#)

Element <extension_schema>

- External Schema in other than w3c XSD language not addressed
- No place to put the xml namespace of the external schema
- Schema versions not handled well
- No one has attached a w3c schema to a registered profile as an appendix
- Some have used xpath like expressions in the <context> element

Element <description_rules>

- An institution may choose to employ particular rules of description when encoding text within elements and attributes of a METS document (e.g., AACR2 for a MARCXML)
- This element should be used to specify all rules of description to be employed in preparing content for a compliant METS document, and where those rules of description should be employed.
- If a profile specifies no rules of description, it should still contain a <description_rules> element with a single <p> subelement stating that the profile specifies no rules of description for conforming documents.

Element <description_rules>

<description_rules>

<p>Descriptive metadata should follow the ODL Guidelines for descriptive metadata

(<http://www2.odl.ox.ac.uk/guidelines/>), including the ODL's name authority rules enumerated within these (essentially conformance to the Anglo-American Name Authority file with supplementary lists specific to the ODL for names absent from this). The metadata is input into the ODL's webform-based metadata system using qualified Dublin Core fields (described in the above guidelines), but converted to the MODS schema when the METS file is generated.</p>

<p>Any names or titles not taken from an authority list should be formatted according to AACR2 conventions.</p>

</description_rules>

Element <controlled_vocabularies>

- An institution may choose to employ certain controlled vocabularies, such as the Library of Congress Subject Headings or the Getty Thesaurus of Geographic Names, for the content of elements within portions of a METS document. A profile for that institution's METS objects should unambiguously identify any controlled vocabularies which are to be used in preparing a METS document conformant with that profile, as well as indicating the specific elements and/or attributes where those controlled vocabularies should be used, including any use in portions of a METS document encoded using an extension schema.
- If no controlled vocabularies are specified by the profile, it should still contain a <controlled_vocabularies> element with a single <p> subelement stating that no controlled vocabularies are specified by this profile.
- may contain 1 <vocabulary>

Element <vocabulary>

- minOccurs="0" maxOccurs="unbounded"
- may contain | <name> | <maintenance_agency>
<URI> | <values> | <context> | <description>
- contained within | <controlled_vocabularies>

Element <controlled_vocabularies>

- Board Action: create a section on METS web site that lists all these vocabularies with links to them (some are external, others will be extract from submitted METS Profiles)

Element <structural_requirements>

- The structural requirements portion of a METS profile allows an institution to delineate additional restrictions on the structure of a conforming METS document beyond those specified by the METS format itself. It is permissible to specify restrictions on the structure of a conforming METS document which cannot be validated by standard XML validation tools. For example, it would be a permissible restriction to state that master still images within a METS document should be contained within a separate file group from derivative images. Possible issues to address in this section include:

Element <structural_requirements>

- Are there any restrictions on the number of occurrences of elements or attributes set forth in the METS schema beyond those specified by the METS schema itself (e.g., there should only be one occurrence of a dmdSec, every conforming document must include a metsHdr element, etc.)?
- Are there any restrictions on the number of occurrences of elements or attributes set forth in extension schema beyond those specified by those schema?

Element <structural_requirements>

- May extension schema only be used within a particular portion of a METS document (e.g., you may wish to specify that a particular extension schema may be used within a <mdWrap> element within a <techMD> section, but that it should not be used within a <sourceMD> section)?

Element <structural_requirements>

- Should the structural map conform to a particular model? For instance, a profile for monographs might specify that the root <div> element must have a TYPE attribute of "book", that all immediately subsidiary <div>s have a TYPE attribute of "chapter". Alternatively, it might specify that there be a root <div> with a TYPE attribute of "text" with subsidiary <div>s having a TYPE attribute of "page".
- Structural metadata is the heart of a METS document, and those creating profiles should try to be as explicit and precise as possible in specifying how structural maps should be created.

Element <structural_requirements>

- Should document authors include metadata within METS document using mdWrap, or reference it using mdRef? Or are both allowable?
- Should content files be included within a METS document using FContent, or referenced using FLocat? Or are both allowable?
- If a profile specifies no structural requirements, it should still contain this element with a single <p> subelement stating that this profile dictates no structural requirements for conforming METS documents.

Element <structural_requirements>

- There are several subelements within the structural_requirements element, one for the root <mets> element, one for each major division within the METS document format, and a final subelement called multiSection.
- Structural requirements should be listed within the subelement identifying the portion of the METS format to which they pertain (e.g., a specification that documents must use FLocat and not FContent to identify data files should appear in the fileSec subelement within structural_requirements). Requirements which span METS documents sections should appear in the multiSection subelement.

Element <structural_requirements>

- Every subelement within the structural_requirements section is composed of a sequence of individual requirement elements. The requirement element has two attributes: 1. an XML ID attribute, and 2. an IDREFS attributed called RELATEDMAT, which you may use to indicate other portions of the profile document where this particular requirement is relevant. Requirement elements are in turn composed of a sequence of paragraph <p> elements.
- may contain | [<metsRootElement>](#)
- [<metsHdr>](#) | [<dmdSec>](#) | [<amdSec>](#) | [<fileSec>](#) | [<structMap>](#) | [<structLink>](#) | [<behaviorSec>](#)
- [<multiSection>](#)

Element <metsRootElement>

```
<metsRootElement>
```

```
  <requirement ID="ROOT_OBJID" RELATEDMAT="DESCR_ID  
APP1_METS_SAMPLE_1">
```

```
    <head>OBJID</head>
```

<p> As previously described, the OBJID attribute must be the primary, persistent and globally unique identifier for the file. This attribute is required; all METS files which are conformant to this profile must have a persistent and globally unique identifier, unless they are Submission Information Packages that will be assigned an identifier upon ingestion. </p>

<p> Computing systems which process files conformant to this profile must preserve this identifier through any transformations, submissions, disseminations, archiving, or other operations on the file. If a system does reassign a new primary identifier to the METS document, the old identifier must be listed as an altRecordID in the metsHdr. The alternate identifiers must also be recorded in the PRIMARY_REPRESENTATION techMD section.</p>

```
  </requirement>
```

```
  <requirement>
```

```
    <head>LABEL</head>
```

```
  ...
```

Element <metsHdr>

<metsHdr>

<requirement ID="metsHdr1">

<p>METS documents of this profile must contain a metsHdr element.</p>

</requirement>

<requirement ID="metsHdr2">

<p>The <metsHdr> element must include a CREATEDATE attribute value.

It should, but is not required to, include a LASTMODDATE attribute value when this differs from the CREATEDATE value.</p>

</requirement>

<requirement ID="metsHdr3">

<p>The <metsHdr> element must include a child <agent> element identifying the person or institution responsible for creating the METS object.</p>

</requirement>

...

Element <dmdSec>

```
<dmdSec>
```

```
  <requirement ID="dmdSec1">
```

```
    <p>Conforming METS documents may, but need not, contain one or more  
&lt;dmdSec&gt; elements. Each &lt;dmdSec&gt; may in turn contain a &lt;dmdRef&gt; or a  
&lt;dmdWrap&gt;. If a &lt;dmdSec&gt; appears with an ID=&quot;dmdSec_fullRecordLink&  
it must have an  
&lt;mdRef&gt; child with an xlink:href attribute containing a URL.  
The METS Navigator application will use this URL to override the  
&quot;linkBackURL&quot; specified in the application configuration  
file.</p>
```

```
  </requirement>
```

```
</dmdSec>
```

Element <amdSec>

<amdSec>

<requirement>

<p>The administrative metadata section contains all technical and digital proven metadata section for the object and its files. There mustn't be any other technical or digital provenance metadata outside of this section.</p>

<p>A conforming METS document must contain at least one techMD section for metadata

on the whole archive object and one techMD section for each file belonging to the object.</p>

<p>The techMD section for the whole archive object must include 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 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.</p>

<p>The respective techMD section of each file must include elements from LMER-File.

Only format with an appropriate attribute REGISTRYNAME to identify the used namespace 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.</p>

...

Element <fileSec>

<fileSec>

<requirement>

<head>General Rules for File Groups and Files</head>

<p> There may be more than one fileGrp element inside the fileSec, and fileGrp elements may be nested. However, similar to the rules for multiple amdSec elements, this profile attaches no meaning to how fileGrp elements are arranged or nested. All linkages between sections are through the file or stream elements and not via the fileGrp elements. This profile essentially treats all file elements as if they were contained inside a single fileGrp. If multiple fileGrp elements are used processors conformant to this profile should preserve them, but this behavior is not guaranteed.</p>

<p> The fileGrp elements must contain a file element for each file which comprises the digital object. </p>

<p> Even though this profile is mostly concerned with files, individual streams within a file such as separate audio streams and video streams in a movie file may be delineated using the stream element if these individual streams have unique structural requirements which are not inherent in the file itself or if

...

Element <structMap>

<structMap>

<requirement ID="structMap1">

<p>A conforming METS document must contain only one <structMap>.</p>

</requirement>

<requirement ID="structMap2" RELATEDMAT="vc2">

<p>A conforming <structMap> must contain a TYPE attribute with the value "physical" or "mixed".</p>

</requirement>

<requirement ID="structMap3">

<p>Each <div> must include a LABEL attribute value.</p>

</requirement>

<requirement ID="structMap4">

<p>A <div> element at any level may point to one or more pertinent <dmdSec> elements via its DMDID attribute value. However, the DMDID attribute should only reference IDs specified at the <dmdSec> element level, and not IDs at lower levels. For example, a <div> DMDID attribute should not reference an ID value of an element within the <xmlData> section of a <dmdSec>.</p>

</requirement>

<requirement ID="structMap5">

<p>A <div> element may or may not directly contain <fptr> elements. (In other words, a <div> of the <structMap> may or may not have content files directly associated with it).</p>

</requirement>

...

</structMap>

Element <structLink>

<structLink>

<requirement>

<head>General Requirements</head>

<p> The structLink element is optional in this profile, and systems which process

files conformant to this profile may ignore this element. However, any structLink elements must be preserved during any operations performed on the files, such as transformations, submissions, disseminations, or archiving. </p>

<p> Even though it is optional and may be ignored by processors, if a structLink element is present it is considered an extension to a given structMap. In other words, a single structLink must be associated with a single structMap: all of the xlink:from and xlink:to attributes contained in a structLink must refer back to xlink:label attributes in the same structMap. However, a given structMap may

...

Element <behaviorSec>

```
<behaviorSec>  
  <requirement ID="behaviorSec1">  
    <p>This profile stipulates no requirements for the  
&lt;behaviorSec&gt;  
element.</p>  
  </requirement>  
</behaviorSec>
```

Element <multiSection>

```
<multiSection>
```

```
  <requirement ID="multi1">
```

<p>Only <file> elements will reference <techMD>, <sourceMD> and <digiprovMD> elements.

In other words, documents implementing this profile will express technical, source, and digital provenance administrative metadata in conjunction with content files only rather than in conjunction with <div> elements in the <structMap>. <rightsMD> elements, however, may be referenced only from <div> elements in the <structMap>.</p>

```
  </requirement>
```

```
  <requirement ID="multi2">
```

<p>Only <div> elements will reference <dmdSec> elements. In other words, documents implementing this profile will express descriptive metadata in conjunction with divisions of the <structMap> and not in conjunction with individual content files (<file> elements).</p>

```
  </requirement>
```

```
  ...
```

```
  </requirement>
```

```
</multiSection>
```

```
</structural requirements>
```

Element <technical_requirements>

- A METS document may reference a variety of external files, including the content files for the METS object (via <FLocat> elements), executable behaviors (via the <mechanism> element), and external metadata files (via <mdRef> elements).
- Institutions may wish to place restrictions on the types of files which may be referenced, such as insisting that all image files be in the TIFF 6.0 format and have a bit-depth between 16 and 32 bits, or that references to external metadata identified as being of type "MARC" via the MDTYPE attribute will point to MARC records conforming to the MARC 21 standard (or alternatively, to an HTML display of a MARC 21 record).

Element <technical_requirements>

- The Technical Requirements section of a profile allows institutions to set forth the full set of restrictions on the technical nature of files which may be referenced from a conformant METS document.
- It should be subdivided into sections for restrictions on content files, restrictions on behavior files, and restrictions on external metadata files. Profile authors should bear in mind that one of the primary purposes of the Technical Requirements section is to allow software developers to anticipate what types of content will be accessible via links from the METS objects, and hence what software is needed to process that content.
- If a profile specifies no technical requirements, it should still contain <technical_requirements> section and the three major subsections, each specifying that the profile imposes no requirements on conforming documents.
- may contain | [<content_files>](#) | [<behavior_files>](#) | [<metadata_files>](#)

Element <technical_requirements>

<content_files>

<requirement>

<p>**This profile supports only image content files.**</p>

</requirement>

<requirement>

<p>**The master (archive) images must be represented and of TIFF format.**</p>

</requirement>

<requirement>

<p>**At least one version of the image content must be of JPEG or GIF format. In other words, at least one content file format must be natively supported by typical internet browsers.**</p>

</requirement>

</content_files>

<metadata_files>

<requirement>

<p>**It is not allowed to reference metadata files. All metadata must be inline.**</p>

</requirement>

</metadata_files>

Element <tool>

- A profile should provide a description of any affiliated tools including validators, stylesheets, authoring tools, rendering applications, which can or should be used with METS documents conforming to the profile. The description should provide a name, description, and URI for each tool. If there are no associated tools, the profile should still contain a single <tool> element with a single <note> subelement stating there are no associated tools for conforming documents.
- may contain | <name> | <agency> | <URI> | <description> | <note>

Element <tool>

<tool>

<name>**kopal Library for Retrieval and Ingest (koLibRI)**</name>

<agency>**Die Deutsche Bibliothek / Staats- und Universitätsbibliothek
Göttingen**</agency>

<URI>**http://kopal.langzeitarchivierung.de/index_koLibRI.php.en**</URI>

<description>

<p>**The kopal Library for Retrival and Ingest (koLibRI) represents a library of Java tools that have been developed for the interaction with DIAS system of IBM within the kopal project. It has been design by intention to be re-usable as a whole or in parts within other contexts, too.**</p>

</description>

</tool>

Element <Appendix>

- A METS profile may contain one or more appendices.
- Every profile must contain at least one appendix containing an example METS document which conforms to the profile, and this example document should always be contained in the first appendix to the profile.

Profile complaints

- They should really be machine actionable
- If they are documents, they need better formatting that `<p>`
- What happens when metadata schemas get upgraded? There needs to be some better versioning mechanism?
- URI is not assigned until after registration
- Not modular (might want to reuse structure, but don't like the descriptive standards)

Machine Actionable

- No programming required, just drop it in to you application and presto, it works!
- Validation
- Display
- Extract metadata, index contents
- Create / export METS matching the profile
- transform METS from one profile to another
- (google it)

Profile Driver

- At CDL, we manage are using over 20 different METS profiles with out machine actionable profiles (and w/o even having a profile document for some profiles).
- an XSLT is written/reused/modified to extract a dublin core record
- an XSLT is written/reused/modified to display
- an XSLT is written/reused/modified to produce javascript on demand
- <http://content.cdlib.org/mets-support/>