Guidelines for using PREMIS with METS for exchange

Revised September 17, 2008

This document offers guidance for using the PREMIS schema(s) as METS extensions. It is intended to suggest common practices for encoding METS documents with PREMIS metadata for exchange purposes. Thus, it is intended for Submission Information Packages or Dissemination Information Packages. For Archival Information Packages, its usage is optional, since some implementations may not choose to store their objects locally this way. Some judgment is needed in considering the purposes for the METS document. Whether the METS document is an exchange object intended for display and delivery may require different encoding decisions than as an object of preservation. When using METS as a submission information package (SIP) a more liberal approach may be warranted, since systems may generate needed data elements, while using it as a dissemination package may require a more restrictive approach and more authoritative metadata.

Background: PREMIS and METS schemas

PREMIS. PREMIS version 1 was implemented in XML schema as 5 separate schemas that reflect the PREMIS data model. In PREMIS version 2 it is implemented as one schema with first level elements for each entity in the data model.

The element <object> in version 2 (and the schema *Object.xsd* in version 1) includes data elements contained in the PREMIS Data Dictionary under the Object entity. Object aggregates information about a digital object held by a preservation repository and describes those characteristics relevant to preservation management. Those characteristics are properties of the Object, which can be at the level of a representation (set of files needed to provide a complete and accurate rendition of an intellectual entity), file, or bitstream.

The element <event> in version 2 (and the schema *Event.xsd* in version 1) includes data elements contained in the PREMIS Data Dictionary under the Event entity. Event aggregates information about an action that involves one or more Objects.

The element <agent> in version 2 (and the schema *Agent.xsd* in version 1) includes data elements contained in the PREMIS Data Dictionary under the Agent entity. Agent aggregates information about attributes or characteristics of agents (persons, organizations, software) associated with rights management and preservation events in the life of an data object.

The element <rights> in version 2 (and the schema *Rights.xsd* in version 1) includes data elements that are related to statements of rights and permissions. Rights are entitlements allowed to agents by copyright or intellectual property law. Permissions are powers or privileges granted by rights holders to other parties.

The element <premis> in version 2 (and the schema *PREMIS.xsd* in version 1) provides for a container element that may be used to keep all PREMIS metadata together.

METS. The METS schema specifies an administrative metadata section (*amdSec*) with the

following subelements:
techMD
rightsMD
sourceMD
digiProvMD

Guidelines for using PREMIS with METS

1. Using PREMIS in METS sections

If not keeping all PREMIS metadata together in the same METS section under amdSec, PREMIS first level data elements should be used in the METS sections as follows.

premis:object under techMD or digiProvMD
Object metadata for file and bitstream levels should be in techMD if separating between METS sections. Implementations may wish to include information about representations in the object schema under digiProvMD, but this is a local decision which may depend on the processing model used. The important consideration is that the structMap and fileSec allow the user to understand what files are being referenced and what metadata apply to them. (see also below under linking).

premis;event under digiProvMD
premis:rights under rightsMD

premis:agent under either digiProvMD (if given in the context of an event) or rightsMD (if given in the context of a rights statement).

If using all PREMIS units together the entire package goes in digiProvMD with the <premis> element as a container.

2. Number of sections to use

Use of one amdSec with repeating subelements (techMD, etc.) or repeating amdSec for each METS subelement is an implementer's preference. These are semantically equivalent provided that the sections are referenced appropriately within the METS document from the fileSec. If referencing the amdSec, you are also referencing the children (i.e. it is a shortcut to referencing all the children). Thus, it is not mandated whether or not to repeat amdSec.

If using premis:agent in conjunction with an event or right, it should be given in its own digiProv or rightsMD section because that minimizes the redundancy (since the same agent may be involved in various events and rights statements). Technical metadata from different schemas (one of which is from PREMIS) may be given in separate techMD sections, or format specific metadata may be included within PREMIS metadata under objectCharacteristicsExtension in premis:object (version 2).

3. Use of PREMIS container

If distributing PREMIS metadata among METS administrative subsections, do not use the PREMIS container. If an implementation wants to keep all PREMIS metadata together the PREMIS container is used and the PREMIS package is in digiProvMD.

4. Redundancies between PREMIS and METS

Redundancies may occur when a format specific technical metadata schema (such as MIX for digital still images) includes data elements that are also in PREMIS. In METS the data from the format specific schema may be used in a separate techMD section with the metadata type indicated using the MDTYPE or OTHERMDTYPE attribute. Note that format specific metadata may also occur in PREMIS version 2 under objectCharacteristicsExtension, which can contain data from external technical metadata schemas.

There are also redundancies when elements are defined both as PREMIS elements and in the METS schema (generally as attributes). Examples are:
SIZE: in PREMIS included in <size> under <objectCharacteristics>; in METS as an attribute of <file> in the <fileGrp> (and also as an element in MIX)
CHECKSUM and CHECKUMTYPE: in PREMIS included in <fixity> under <objectCharacteristics>; in METS as attributes of <file> (and also as an element in MIX)
MIMETYPE: in PREMIS included in <format> under <objectCharacteristics>; in METS as an attribute of <file> (also as an element in MIX).

An application may decide whether it is easier to include the information redundantly, based on how the data will be used and/or supplied. Implementers should consider the use of the metadata (e.g. display or preservation) and whether the METS (for display) or PREMIS (for preservation) is primary when deciding which to use and whether to record redundantly. Another consideration is which schema is more expressive. In most cases (except for structural metadata in METS vs. PREMIS) the PREMIS element is more detailed (e.g. format). When the use of the metadata is not clear or display and preservation are of equal importance, err on the side of redundancy and include both PREMIS and METS constructs. Implementers should document how their application handles redundancies in a profile.

5. METS structMap and PREMIS structural relationship elements

Hierarchical structural relationships should be detailed as nested <div> elements according to the METS schema and rules because it is richer than that provided as PREMIS semantic units. If the scope of exchange objects is preservation, implementers should also use the PREMIS relationship elements in the Object schema for structural relationships. PREMIS relationship elements should always be used for derivative types of relationships.

6. METS ID/IDREF and PREMIS identifier elements

There are several ways to make linkages between elements using PREMIS and METS. There are separate data elements in PREMIS for identifiers: objectIdentifier, eventIdentifier, agentIdentifier, and rightsStatementIdentifier (in version 2; called permissionStatementIdentifier in version 1). There are also linking identifiers for linking from one PREMIS entity to another.

In addition, ID/IDREF constructs have been provided in the PREMIS schemas that provide linkages between PREMIS related elements (RelObjectXmlID, LinkEventXmlID, LinkPermissionStatementXmlID, RelEventXmlID, linkingObjectXmlID, LinkAgentXmlID, GrantAgentXmlID). METS also provides ID/IDrefs that link between files and their related metadata in the appropriate sections.

Implementers need to rely on METS ID/IDRefs when referencing a METS section (e.g. techMD) from a file in the fileSec when mdWrap is used. Links should be made at the highest level possible, usually pointing to the first level subelement under amdSec. Linkages between metadata sections for different PREMIS entity metadata using the PREMIS first level elements in version 2.0 (or the different PREMIS schemas in version 1.1) will also use the METS ID/IDref constructs, rather than the PREMIS ID/IDRefs. Implementers should also include the explicit PREMIS linking identifier elements (e.g. linkingEventIdentifier in Object; linkingAgentIdentifier in Rights) in the event that the PREMIS metadata is used as a whole outside of the METS document. When PREMIS metadata is outside of the METS container, ID/IDRefs may break, so PREMIS linking identifier elements are needed. Linking using PREMIS constructs may provide more detailed semantics than the METS linking itself. In these cases two-way linking (i.e. using both METS ID/IDRefs and PREMIS identifier elements) may be necessary.

When establishing the values of METS IDs, it is important that implementers create appropriate IDs by assuring that they are unique within the instance, maintain their persistence, and that everything is referenced that needs to be. There should not be any orphaned elements, e.g. a techMD that isn't referenced in the structMap/fileSec. Any sections with PREMIS elements should be referenced; orphaned elements will be ignored. Note that these rules hold for METS in general with or without PREMIS as an extension schema.

7. Use of METS profiles

Implementations are encouraged to establish profiles that specify options chosen for particular applications or where this document is not prescriptive and, where there are redundancies, state whether elements in PREMIS or METS are primary. Profiles may also suggest policy for resolving conflicts between metadata included redundantly using both PREMIS and METS elements or PREMIS and format-specific elements.

A generic METS profile incorporating PREMIS schemas as METS extension schemas will be provided and may be used if desired as a basis for particular applications.

8. Example of PREMIS 2.0 in a METS document

An example document using this profile may be found at:
http://www.loc.gov/standards/premis/louis-2-0.xml

In this document the following choices were made:
PREMIS object is in techMD; event is in digiProv.
There is one amdSec with multiple techMD, digiProv, etc.

MIX metadata is under premis:objectCharacteristicsExtension in the appropriate techMD section.

Where there are redundancies, the premis elements generally are used. For instance, fixity is in PREMIS, not as attributes in the METS fileGrp. MIMETYPE has been retained in the fileGrp, but is also in <format> under techMD/object. The equivalent MIX elements that have PREMIS equivalents are not used.

PREMIS relationships are used for both structural and derivative relationships, even though structural relationships are also detailed in the METS structMap.

PREMIS identifiers are used as well as METS ID/IDRef constructs.

9. Acknowledgements

The following individuals participated in the development of these guidelines:

Rebecca Guenther, Library of Congress (PREMIS Editorial Committee chair)
Rob Wolfe, MIT (METS liaison)
Olaf Brandt, Royal Library of the Netherlands
Markus Enders, British Library
Tom Habing, University of Illinois Urbana/Champaign
Francesco Lazzarino, Florida Center for Library Automation
Clay Redding, Library of Congress
Jenn Riley, Indiana University