PREMIS PREservation Metadata Implementation Strategies

# AGENTS, RIGHTS, EVENTS

## Agents

- The Agent entity aggregates information about agents (persons, organizations or software) associated with rights management and/or preservation events in the life of an object.
- Intended only to identify the agent unambiguously, and to allow linking from other entity types.
- Repositories encouraged to use any richer scheme that may be appropriate.

agentIdentifier (mandatory)
    agentIdentifierType (mandatory)
    agentIdentifierValue (mandatory)
agentName (optional)
agentType (optional)
agentNote (optional) **COMING**
agentExtension (optional) **COMING**

# Examples of agents

agentIdentifier
    agentIdentifierType = lcnaf
    agentIdentifierValue = oca05896076
agentName = Caplan, Priscilla
agentType = person

agentIdentifier
    agentIdentifierType = repositoryX
    agentIdentifierValue = 57
agentName = Timberline Publishing Company
agentType = organization

agentIdentifier
    agentIdentifierType = fda
    agentIdentifierValue = daitss1.4.14
agentName =
agentType = software

## Rights

- The Rights entity aggregates information about statements of rights and permissions
- PREMIS version 1 addressed only narrow scope: what *permissions* have been granted to the repository itself to carry out actions related to objects within the repository
- PREMIS v 2.0 was extended and internationalized
  - Enabled for different legislations
  - Not bound to contracts alone, extended for different law and entitlement systems (e.g. moral rights or Urheberrecht)
  - Distinguishes between rights granted by copyright law, statute or license agreement
  - Extension for other right expressions added

## High level semantic units Rights

rightsStatement (optional)

- rightsStatementIdentifier (mandatory)
- rightsBasis (mandatory)
- copyrightInformation (optional)
- licenseInformation (optional)
- statuteInformation (optional)
- rightsGranted (optional)
- linkingObjectIdentifier (optional)
- linkingAgentIdentifier (optional)

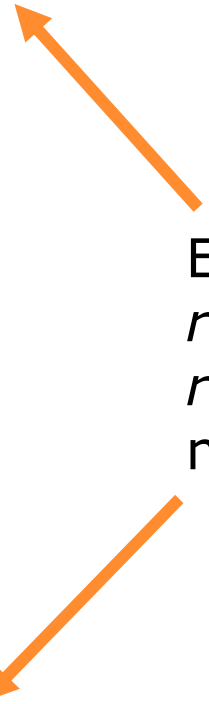
rightsExtension (optional)

## High level semantic units Rights

rightsStatement (optional)

- rightsStatementIdentifier (mandatory)
- rightsBasis (mandatory)
- copyrightInformation (optional)
- licenseInformation (optional)
- statuteInformation (optional)
- rightsGranted (optional)
- linkingObjectIdentifier (optional)
- linkingAgentIdentifier (optional)

rightsExtension (optional)



Either *rightsStatement* or *rightsExtension* must be present!

# rightsBasis

- Designation of the basis for the right or permission
- Values should be taken from a *controlled vocabulary*
- Values: copyright, license, statute.
  - If *rightsBasis* is
  - "copyright", *copyrightInformation* should be provided.
  - "license", *licenseInformation* should be provided.
  - "statute", *statuteInformation* should be provided.

- If more than one basis applies, the entire rights entity should be *repeated*.

# Example rightsBasis and statuteInformation

rightsBasis = statute

statuteInformation

- statuteJurisdiction = de (ISO 3166)
- statuteCitation = „Gesetz über die deutsche Nationalbibliothek vom 22. Juni 2006(DNBG)“
- statuteInformationDeterminationDate = 20080901
- statuteNote = Legal Deposit Law in Germany (Applicability to web-published content)

rightsStatement

- rightsStatementIdentifier
- rightsBasis
- copyrightInformation
- licenseInformation
- **statuteInformation**
- rightsGranted
- linkingObjectIdentifier
- linkingAgentIdentifier

# Example rightsBasis and copyrightInformation

rightsStatement
- rightsStatementIdentifier
- rightsBasis
- **copyrightInformation**
- licenseInformation
- statuteInformation
- rightsGranted
- linkingObjectIdentifier
- linkingAgentIdentifier

rightsBasis = copyright

copyrightInformation

- copyrightStatus = copyrighted
- copyrightJurisdiction = us (taken from ISO 3166)
- copyrightStatusDeterminationDate = 20080910
- copyrightNote = Copyright expiration expected in 2022

# Example rightsBasis and licenseInformation

rightsStatement
- rightsStatementIdentifier
- rightsBasis
- copyrightInformation
- **licenseInformation**
- statuteInformation
- rightsGranted
- linkingObjectIdentifier
- linkingAgentIdentifier

rightsBasis = license

licenseInformation

licenseIdentifier

licenseIdentifierType = fda

licenseIdentifierValue = 3954

licenseTerms = The actual license text and other license related information…

licenseNote = „License embedded in XMP block in file header”

# rightsGranted

- rightsGranted (optional)
  - act (mandatory)
  - restriction (optional)
  - termOfGrant (mandatory)
    - startDate (mandatory)
    - endDate (optional)
  - rightsGrantedNote (optional)

- rightsStatement
  - rightsStatementIdentifier
  - rightsBasis
  - copyrightInformation
  - licenseInformation
  - statuteInformation
  - **rightsGranted**
  - linkingObjectIdentifier
  - linkingAgentIdentifier

# rightsGranted.act

- **The action the repository is granted permission to take**
- Some suggested values:
  - replicate = make an exact copy
  - migrate = make a copy identical in content in a different file format
  - modify = make a version different in content
  - use = read without copying or modifying (e.g., to validate a file or run a program)
  - disseminate = create a DIP for use outside of the preservation repository
  - delete = remove from the repository
- Granularity up to the preserving institution
- Useful to employ the same controlled values as in *eventType*.

## rightsGranted.restriction

- A condition or limitation on rightsGranted.act
- For example:
  - act = replicate
  - restriction = no more than 3 copies at any time
  - act = disseminate
  - restriction = rightsholder must be notified after the fact
- Repeatable if there are multiple conditions/limitations

# rightsGranted.termOfGrant

- Beginning and ending dates of rights granted
- Extended Date Time Format in schema, which is compliant with ISO 8601 format (with extensions)

http://www.loc.gov/standards/datetime

- Examples

termOfGrant
startDate = 20050101
endDate = 20150101

termOfGrant
startDate = 1900
endDate = OPEN

## rightsGranted.rightsGrantedNote

- Defined as additional information about the rights granted
- Statement about risk assessment, for example, when a repository is not certain about what permissions have been granted
- Examples:
  - no contact information for rightsholder found
  - Unclear: restriction on dissemination possible

# Other rightsStatement information

- rightsStatementIdentifier (mandatory)
  - rightsStatementIdentifierType (mandatory)
  - rightsStatementIdentifierValue (mandatory)
- linkingObjectIdentifier (optional)
- linkingAgentIdentifier (optional)

# rightsExtension

- Container to include semantic units defined outside of PREMIS
- To replace or extend PREMIS (instead of or in addition)
- If you are using an extension schema, a reference to that schema must be provided!
- If *rightsExtension* container needs to be associated explicitly with any PREMIS subunit under *rights*, the container *rights* is repeated.
- If extensions from different external schemas are needed, *rights* should also be repeated.
  - E.g. with California Digital Library's copyrightMD schema www.cdlib.org/inside/projects/rights/schema/

# Events

- The Event entity aggregates information about an action involving one or more Objects
- Recording events can be very important
  - to demonstrate digital provenance
  - to prove that rights have not been violated
  - as an audit trail
  - for problem solving if something goes wrong
  - for billing or reporting
- Judgement calls
  - what exactly are the boundaries of an Event?
  - what actions are worth recording as Events?

# High level semantic units

- eventIdentifier (mandatory)
- eventType (mandatory)
- eventDateTime (mandatory)
- eventDetail (optional)
- eventOutcomeInformation (optional)
- linkingAgentIdentifier (optional)
- linkingObjectIdentifier (optional)

# eventType

- Names the event
- From a controlled vocabulary
- Could use coded values
- Granularity is implementation-specific

| | |
|---|---|
| Capture | Compression |
| Deaccession | Decompression |
| Decryption | Deletion |
| Dig. signature validation | Dissemination |
| Fixity check | Ingestion |
| Message digest calculation | Migration |
| Normalization | Replication |
| Validation | Virus check |

# eventDetail

- **Additional information about the event**
- Not necessarily intended to be machine-processable, but could be structured to allow this
- For example:

eventType = dissemination

eventDetail = A001923;WS;20060413T071530-0500

[the agent requesting the dissemination; a dissemination type code; and the date/time of the request (which could be different from the time of the actual dissemination itself)]

# eventOutcomeInformation

- Structure

eventOutcomeInformation
- eventOutcome = intended to be coded
- eventOutcomeDetail = more granular information
  - eventOutcomeDetailNote = textual form
  - eventOutcomeDetailExtension = container to include semantic units defined outside of PREMIS

# eventDetail

Structure:

eventOutcomeInformation
- eventOutcome
- eventOutcomeDetail (optional)
  - eventOutcomeDetailNote
  - eventOutcomeDetailExtension

Either *eventOutcomeDetailNote* or *eventOutcomeDetailExtension* must be present, if *eventOutcomeDetail* is used!

# Examples for eventOutcomeInformation

eventOutcomeInformation
  eventOutcome = 00 [means ok]
    eventOutcomeDetail
    eventOutcomeDetailNote = new file successfully created

eventOutcomeInformation
    eventOutcome = FV-S
        [means file validation successful]
    eventOutcomeDetail
      eventOutcomeDetailNote = A4,A14,A19
        [coded list of validation errors found]

# linking Events with Agents and Objects

- linkingAgentIdentifier
  - linkingAgentIdentifierType
  - linkingAgentIdentifierValue
  - linkingAgentRole = because the same Agent may have different Role in the digital Archive system
- linkingObjectIdentifier
  - linkingObjectIdentifierType
  - linkingObjectIdentifierValue
  - linkingObjectRole

## Event Example:

The TIFF FILES described in the Objects example were deposited into the LC digital repository on July 6, 2006.

This process consists of two Events: validation and ingest.

According to the repository's business rules, the repository itself is considered the Agent of record for all Events.

Therefore, software used to carry out Event processes is recorded in eventDetail.

# First Event: Validation

| | |
|---|---|
| **eventIdentifier** | |
| **eventIdentifierType** | LocalRepository |
| **eventIdentifierValue** | E001.1 |
| **eventType** | Validation |
| **eventDateTime** | 2006-07-06T01:05:07.001 |
| **eventDetail** | jhove_1e |
| **eventOutcomeInformation** | |
| **eventOutcome** | successful |
| **eventOutcomeDetail** | |
| **eventOutcomeDetailNote** | well-formed and valid |
| **linkingAgentIdentifier** | |
| **linkingAgentIdentifierType** | AgentID |
| **linkingAgentIdentifierValue** | na12345 |
| **linkingAgentIdentifierRole** | Authorizer |
| **linkingObjectIdentifier** | |
| **linkingObjectIdentifierType** | hdl |
| **linkingObjectIdentifierValue** | loc.music/gottlieb.09601 |
| **linkingObjectRole** | input |

# Second Event: Ingest

| | |
|---|---|
| **eventIdentifier** | |
| **eventIdentifierType** | LocalRepository |
| **eventIdentifierValue** | E001.2 |
| **eventType** | Ingest |
| **eventDateTime** | 2006-07-06T07:07:23.002 |
| **eventDetail** | ingester1_0.exe |
| **eventOutcomeInformation** | |
| **eventOutcome** | successful |
| **eventOutcomeDetail** | |
| **eventOutcomeDetailNote** | Object transferred to storage |
| **linkingAgentIdentifier** | |
| **linkingAgentIdentifierType** | AgentID |
| **linkingAgentIdentifierValue** | na12345 |
| **linkingAgentIdentifierRole** | Authorizer |
| **linkingObjectIdentifier** | |
| **linkingObjectIdentifierType** | hdl |
| **linkingObjectIdentifierValue** | loc.music/gottlieb.09601 |
| **linkingObjectRole** | source |