Rosetta DNX Profile

The DNX schema is a simple and unified XML schema that holds the administrative metadata of the IE in the permanent repository. It contains all the important data elements in a simple flat structure, divided between the different object levels (IE, representation, file and bitstreams), and includes the important technical metadata that is relevant for preservation.

The administrative metadata that needs to be stored arrives from various sources:

- Technical metadata that is being generated by the metadata extraction tools (JHOVE, NLNZ tools)
- Access rights associated with the material flow
- CMS information (system and record ID)
- Provenance information – Producer, Producer Agent information, events information
- Structural IE relationships – provided by the depositing or editing users
- Miscellaneous information – such as links to external events, or other intellectual entities

Since all this information comes from different sources with different standards, some of it is duplicated or organized in a way that is not useful. The DNX profile, therefore, is designed to hold all this information in a clear and organized way, with a clear mapping to the original source that enables converting it back and forth.

The DNX is written to the AIP (METS XML file) based on the metadata that is stored in different tables in the Rosetta staging database. Most of the DNX data is generated by Rosetta, while some of the data in the DNX section is populated by the submission application, before the IE is deposited.

The provenance information is written in the DNX when the data is moved to the permanent stage, since the information is still gathered during the SIP processing stage.

The purpose of this document is to describe the DNX profile. This document includes all the information about the sections and elements of the DNX schema, such as the description of each field, the data source of the field, the matching PREMIS semantic unit, and the phase at which the IE lifecycle is created.

DNX and PREMIS

Most of the DNX sections and fields come from the PREMIS data dictionary. Rosetta implements PREMIS (PREMIS compliant), and most of the PREMIS semantic units are represented in the DNX profile. In case semantic units will be added to PREMIS, they will considerably be added to the DNX profile.

Note

Note: Not all the PREMIS fields in the DNX are managed automatically by Rosetta. Some fields can only be filled in and monitored manually – for example, the fields that hold the relationships between different IEs (relationship DNX section).

The differences between the PREMIS data model and Rosetta's data model is that in Permis, the **Agents** entity holds the details of an agent, which is a person, organization, or software program/system associated with events in the life of an object, or with rights attached to an object. In Rosetta, the agent is only an attribute of an external provenance event, since in the other areas, Rosetta is the agent associated with events in the life of the objects and the access rights attached to

the IE.

DNX Section Structure

The DNX format is built from logical groups of metadata fields called Sections.

Each DNX section contains a group of fields that are related to each other. For example, the section **generalRepCharacteristics** (General Representation Characteristics) includes the fields that describe the parameters of the representation – Preservation Type, Usage Type, Revision Number, and so forth.

Most of the sections come from the PREMIS data dictionary, but some of them are unique to Rosetta. The structure of a DNX section is as follows:

```
<section id=" Section Name ">
<record>
 <key id="Field Name">Field Value</key>
…
</record>
</section>
```

Each record holds the fields of the section in the form of:

```
<key id=Name>Value</key>
```

The following example illustrates this:

```
- <dnx xmlns="http://www.exlibrisgroup.com/dps/dnx">
  - <section id="generalRepCharacteristics">
    - <record>
        <key id="preservationType">PRESERVATION_MASTER</key>
        <key id="usageType">VIEW</key>
        <key id="RevisionNumber">1</key>
        <key id="DigitalOriginal">false</key>
      </record>
    </section>
```

Structure of a Repeatable Section

If a DNX section is repeatable, there will be multiple records of the same structure, as shown in the following example:

```
  - <section id="internalIdentifier">
    - <record>
        <key id="internalIdentifierType">SIPID</key>
        <key id="internalIdentifierValue">20</key>
      </record>
    - <record>
        <key id="internalIdentifierType">PID</key>
        <key id="internalIdentifierValue">REP1101</key>
      </record>
    - <record>
        <key id="internalIdentifierType">DepositSetID</key>
        <key id="internalIdentifierValue">41</key>
      </record>
    </section>
```

Events within DNX

The event metadata holds the information about actions that affect the object. Each object level has different types of actions that should be captured. In Rosetta, the events that are recorded in the AIP are provenance events, while many

other events are captured in the system but do not become part of the AIP metadata.

All events that are generated by the system are written to a database table. Events that are indicated as provenance (in the code, non-configurable) are copied from the table of events to the METS file, while the non-provenance events remain in the table.

The storage of events in a table allows the creation of reports that show the statistics regarding various activities.

Provenance Events

The following types of events are considered provenance events:

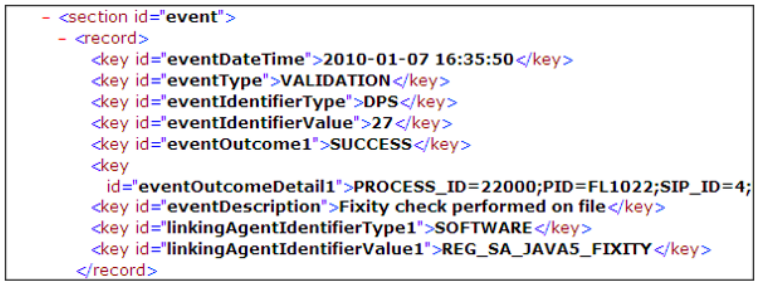
- Changes to the IE metadata – adding metadata to any of the IE levels (descriptive DC, source MD, access rights policy, structural map, DNX)
- Addition of a new Representation – new Representation that was added through the Web Editor or as a result of a Preservation Action
- Validation checks – validity and integrity checks on files (Note – Fixity check will not generate a provenance event unless calculated fixity is different than the previous one)
- Enrichment – generation of a persistent identifier

Each such event will be written in the events (mets:digiprovMD) section belonging to the relevant object level (IE, representation, or file).

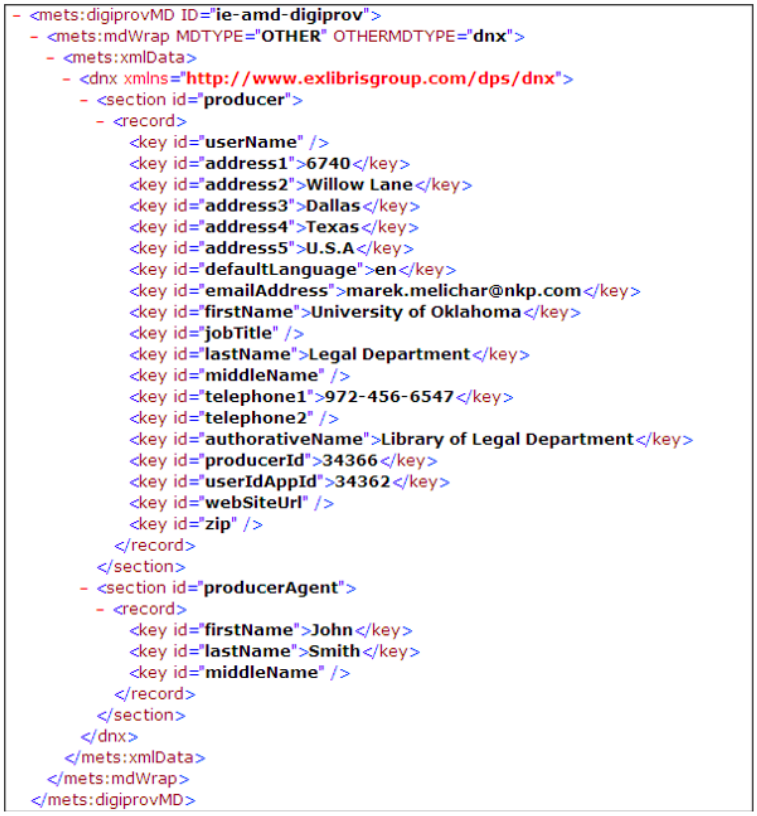
Each event will be written in the DNX format and will include the following:

- Agent – The agent that triggered this event. An agent is not necessarily a person. An agent may also refer to a process, plug-in tool, and so forth
- Event details – Such as the creation date, a description, the parameters, and so forth

Following is an example of an event that is stored in the `digiProvMD` section of a file. This section holds the events in DNX format:



In addition to events, the `digiprovMD` section on the IE level stores the details of the Producer and the Producer Agent who deposited the IE. This section is populated automatically for each IE in Rosetta and includes all the information of the Producer as it exists in Rosetta at the time of the deposit:



Access Rights Within DNX

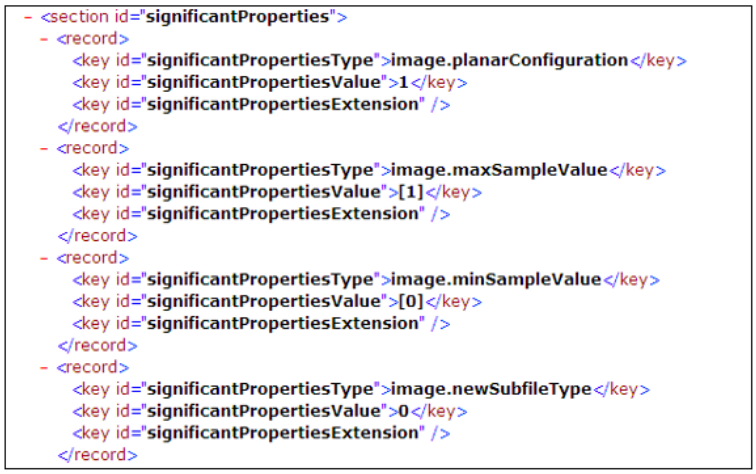
Two types of rights are stored in the DNX sections: PREMIS and non-PREMIS.

- PREMIS rights (IE only) – Information regarding an external system that manages the IE's rights. Note that these rights are not mandatory, and they are not managed or enforced by Rosetta. There is one DNX section for holding the details of these rights:
- `linkingRightsStatementIdentifier` – Holds the type and the value of the statement identifier, if it is generated and stored in a repository other than Rosetta.
  - `linkingRightsStatementIdentifierType` – A designation of the domain within which the `linkingRightsStatementIdentifier` is unique
  - `linkingRightsStatementIdentifierValue` – The value of the `linkingRightsStatementIdentifier`
- Non PREMIS (IE, Representation, and File) – Information regarding the access rights policy managed by Rosetta. Note that it is mandatory for each IE to have an associated access rights policy, while for representation and file access rights are optional. The DNX section for holding this information is `accessRightsPolicy`. The following fields are part of this section:
- PolicyID – The unique ID of the different access rights managed by Rosetta. For example: AR_EMBARGOED_FOR_5_YEARS, AR_5_CONCURRENT_USERS
- Policy description – Description of the policyID. For example: AR_EMBARGOED_FOR_5_YEARS – Embargoed for 5 years, AR_5_CONCURRENT_USERS – Limited access according to copyright law
- Policy parameters – If the policy requires any parameters

Significant Properties of Files Within DNX

To have a scalable structure that supports additions of technical metadata over the years, the DNX section that holds the

extracted technical metadata for each file has the following structure:



This structure allows defining the technical attributes as the values of the `significantPropertiesType` fields, and their values as the values of the `significantPropertiesValue` fields.

DNX Sections

Below is the description for each of the DNX sections.

- IE level
  - generalIECharacteristics
  - objectCharacteristics
  - CMS
  - webHarvesting
  - internalIdentifier
  - objectIdentifier
  - significantProperties
  - linkingIEIdentifier
  - producer
  - producerAgent
  - event
  - linkingRightsStatementIdentifier
  - accessRightsPolicy
  - grantedRightsStatement
  - metadata
  - collection
  - retentionPeriodPolicy
  - IERelationship

- Representation level
  - generalRepCharacteristics
  - objectCharacteristics

- internalIdentifier
- significantProperties
- linkingIEIdentifier
- event
- linkingRightsStatementIdentifier
- metadata
- preservationLevel
- environment
- environmentDependencies
- environmentSoftware
- envSoftwareRegistry
- environmentHardware
- envHardwareRegistry
- environmentExtension
- relationship

- File level
  - generalRepCharacteristics
  - objectCharacteristics
  - internalIdentifier
  - significantProperties
  - linkingIEIdentifier
  - event
  - linkingRightsStatementIdentifier
  - metadata
  - preservationLevel
  - environment
  - environmentDependencies
  - environmentSoftware
  - envSoftwareRegistry
  - environmentHardware
  - envHardwareRegistry
  - environmentExtension
  - relationship
  - fileFixity
  - fileFormat
  - fileVirusCheck
  - fileValidation
  - fileTechnicalMetadataExtraction
  - vsOutcome
  - creatingApplication
  - Inhibitors
  - objectCharacteristicsExtension

- signatureInformation
- signatureInformationExtension

- BitStream level
  - generalRepCharacteristics
  - significantProperties

Below is the description for each of the DNX sections

Note

Defining a section as **Mandatory** means that the information stored in the section is required by Rosetta for its functioning. For example, without the internal identifier, objects cannot be searched and found and without populated Format ID, Rosetta is not able to perform any preservation activities.

This is not the meaning of 'Mandatory' according to PREMIS, and there is no contradiction between the two definitions – Rosetta allows its users to define which fields must be populated as part of the SIP processing. For more details regarding metadata validation, see the Rosetta Configuration Guide.

General IE/Rep/File Characteristics

Definition	The **generalIECharacteristics**, **generalRepCharacteristics**, **generalFileCharacteristics** sections contain administrative as well as control attributes that determine how objects are delivered, published, and searched.
Rosetta Mandatory	Yes – Not every field
Source	User
Repeatable	No
Level	IE, Representation, File and BitStream
METS section	techMD

(Rosetta) Object Characteristics

Definition	**objectCharacteristics** – This section can be on each level (IE, representation, and file) and it contains control attributes that are relevant on all levels, such as dates and user information.
Rosetta Mandatory	Yes
Source	System/User
Repeatable	No
Level	IE, Representation, File and BitStream
METS section	techMD

cms

Definition	This section holds the Collection Management System details. Each IE in Rosetta can have a “handle” to descriptive metadata that is managed in the ILS, such as Aleph or Voyager. Since this information might be relevant for many IEs and in order to allow a single point of update, the IE holds only the reference to this information, without the need to duplicate it in Rosetta.
Rosetta Mandatory	No
Source	User/System
Repeatable	No
Level	IE
METS section	techMD

Web Harvesting

Definition	**webHarvesting** – This section contains the information regarding Web harvesting. It describes the tool that was used for building the Web archive file and some other parameters of this action. (This section was added because there is no existing set of fields that can hold this metadata according to PREMIS).
Rosetta Mandatory	No
Source	User
Repeatable	No
Level	IE
METS section	techMD

Producer

Definition	This section holds the information of the Producer as it is stored in the staging DB.
Rosetta Mandatory	Yes
Source	System
Repeatable	No
Level	IE
METS section	digiprovMD

Producer Agent

Definition	**producerAgent** – This section holds the information of the Producer Agent who deposited the IE. (It contains only the name, not the entire user record).
Rosetta Mandatory	Yes
Source	System
Repeatable	No
Level	IE
METS section	digiprovMD

Access Rights Policy

Definition	**accessRightsPolicy** – This section holds the access rights policy details that are checked before delivery. The system analyzes whether the calling user is authorized to view the object.
Rosetta Mandatory	Yes
Source	System/User
Repeatable	No
Level	IE, Representation, File
METS section	rightsMD

Granted Rights Statement

Definition	**grantedRightsStatement** – This section holds the copyrights statement that was presented to the Producer Agent upon depositing the IE (boilerplates as part of the material flow). It is currently not in use.
Rosetta Mandatory	No (Currently not in use)
Source	System/User
Repeatable	Yes (no limits)
Level	IE
METS section	rightsMD

Metadata (Deprecated)

Definition	This table is deprecated and not in use. This record holds the details of the HDEMETADATA record that is kept in the sourceMD METS section. The details are used by the system to allow accurate matching between the data in the METS to the data in the DB, when the IE is loaded back to the staging DB from the permanent repository. The details include the ID and the type (DC, DNX_REP, and so forth) as well as the control dates (creation, modification).
Rosetta Mandatory	No
Source	System
Repeatable	Yes (no limits)
Level	IE, Representation and File
METS section	sourceMD

Retention Policy

Definition	Hold the details of Retention Policy ID which determines the duration required to preserve content, after which content will be deleted.
Rosetta Mandatory	No
Source	User
Repeatable	No
Level	IE
METS section	techMD

Internal Identifier

Definition	**internalIdentifier** – This section holds a record for each of the identifiers that are created by Rosetta, such as PID, SIP ID, and Deposit Set ID. Each object level has its own section of identifiers (there is a PID for each IE, representation, and file), while on the IE level there are other identifiers (such as SIP ID).
Rosetta Mandatory	Yes – All types of internal identifiers are Rosetta Mandatory since they are created and used by the system
Source	System
Repeatable	Yes (no limits)
Level	IE, Representation, and File
METS section	techMD

Object Identifier

Definition	**objectIdentifier** – This section holds the identifiers of the IE that are stored in an external system – for example, Handle and URN: NBN. These identifiers are not internal in the sense that in Rosetta they are used only as metadata, and not as identifiers. These identifiers can be generated in Rosetta by a plug-in or they can be populated pre-ingest by the submission application.
Rosetta Mandatory	No
Source	User/System
Repeatable	Yes (no limits)
Level	IE, Representation, and File
METS section	techMD

Preservation Level

Definition	**preservationLevel** – This section holds information indicating the decision or policy on the set of preservation functions to be applied to an IE and the context in which the decision or policy was made.
Rosetta Mandatory	No
Source	User
Repeatable	No
Level	Representation and File
METS section	techMD

Significant Properties

Definition	**significantProperties** – This section holds the extracted technical metadata for each file. However, it can be used in any of the other levels and it can hold other properties that were not extracted by the MD Extraction tool(s).
Rosetta Mandatory	No (Depends on the MD Extraction tool that is associated with the Format)
Source	System/User
Repeatable	Yes (no limits)
Level	IE, Representation File and BitStream
METS section	techMD

File Fixity

Definition	**fileFixity** – For each file, this section holds a record for each checksum algorithm that is used by the validation stack (SHA-1, CRC32, and MD5).
Rosetta Mandatory	No
Source	System
Repeatable	Yes – For every checksum algorithm in use by the Fixity task
Level	File
METS section	techMD

File Format

Definition	**fileFormat** – For each file, this section holds the format details as they were identified by the format identification task in the validation stack.
Rosetta Mandatory	Yes
Source	System/User
Repeatable	Yes
Level	File
METS section	techMD

File Virus Check

Definition	**fileVirusCheck** – For each file, this section holds the results of the virus check that was performed as part of the validation stack.
Rosetta Mandatory	No
Source	System
Repeatable	No
Level	File
METS section	techMD

File Validation

Definition	**fileValidation** – For each file, this section holds the details and the results (valid/invalid, well-formed/not well formed) of the format validation tool that was used by the Format Validation task (or the soon to be deprecated MD Extraction with Validation task) as part of the validation stack. Note that this section does not hold the actual output of the extraction tool (for example, JHOVE). The output is stored in the significant properties section and holds the information about the extraction tool.
Rosetta Mandatory	No
Source	System
Repeatable	No
Level	File
METS section	techMD

File Technical Metadata Extraction

Definition	**fileTechnicalMetadataExtraction** – For each file, this section holds the extraction tool information (agent name, plug-in name, errors when relevant) of the technical MD extraction tool that was used by the MD Extraction task as part of the validation stack. Note that this section does not hold the actual output of the extraction tool (for example, JHOVE). The output is stored in the significant properties section, while this section holds the information about the extraction tool.
Rosetta Mandatory	No
Source	System
Repeatable	No
Level	File
METS section	techMD

Validation Stack Outcome

Definition	**vsOutcome** – This section holds the information about the validation routines that were used to validate the files. The validation includes the following: a virus check, fixity check, format identification, technical metadata extraction and risk extraction. Different plug-ins can be used and their details are captured in this section.
Rosetta Mandatory	Yes
Source	System
Repeatable	Yes – Repeated for every task in the VS task chain
Level	File
METS section	techMD

Creating Application

Definition	**creatingApplication** – For each file, this section holds the information about the application that was used for creating the file, which was created before it was deposited or in Rosetta as part of a preservation action.
Rosetta Mandatory	No
Source	System/User
Repeatable	No
Level	File
METS section	techMD

Inhibitors

Definition	On a file level, this section holds the features intended to inhibit access, use, or migration.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	File
METS section	techMD

Object Characteristics Extension

Definition	**objectCharacteristicsExtension** – On a file level, this is a container for including semantic units that are not DNX.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	File
METS section	techMD

Environment

Definition	On a file or representation level, this section holds the details of hardware/software combination that supports the usage (rendering, viewing) of the representation/file.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	Representation, File
METS section	techMD

Environment Dependencies

Definition	**environmentDependencies** – On a file or representation level, this section holds information about a non-software component or associated file required in order to use or render the representation or file - for example, a schema, DTD, or an entity file declaration.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	Representation, File
METS section	techMD

Environment Software

Definition	**environmentSoftware** – This section holds the details of the software that is needed for rendering the object (file, representation). The details include name, version, type, and dependencies.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	Representation, File
METS section	techMD

Environment Software Registry

Definition	**envSoftwareRegistry** – This section holds the details of the registry in which the environment software is registered.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	Representation, File
METS section	techMD

Environment Hardware

Definition	**environmentHardware** – This section holds the details of the hardware that is required for rendering the object (file, representation). The details include name and type.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	Representation, File
METS section	techMD

Environment Hardware Registry

Definition	**envHardwareRegistry** – This section holds the details of the registry in which the environment hardware is registered.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	Representation, File
METS section	techMD

Environment Extension

Definition	**environmentExtension** – This section is a container for including semantic units that are not DNX.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	Representation, File
METS section	techMD

Signature Information

Definition	**signatureInformation** – On a file level, this section can hold the information that is required for using a digital signature to authenticate the signer of an object and/or the information contained in the object.
Rosetta Mandatory	No
Source	User
Repeatable	No
Level	File
METS section	techMD

Signature Information Extension

Definition	**signatureInformationExtension** – This section holds digital signature information using semantic units that are not DNX.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	File
METS section	techMD

Relationship

Definition	This section holds the relations between files or between representations, if there are any.
Rosetta Mandatory	No
Source	User/System (During Add Representation or Preservation Action)
Repeatable	Yes (no limits)
Level	File, Representation
METS section	techMD

IE Relationship

Definition	This section holds the structural IE relationships between a parent structural IE and its child IEs.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	IE
METS section	techMD

Linking IE Identifier

Definition	**linkingIEIdentifier** – This section holds the identifier of a different IE that is related to the object (IE, representation, or file).
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	IE, Representation or File
METS section	techMD

Event

Definition	This section holds the provenance events on each level (IE, representation, and file).
Rosetta Mandatory	Yes – The provenance events are Rosetta Mandatory.
Source	User/System
Repeatable	Yes (no limits)
Level	IE, Representation or File
METS section	digiprovMD

Linking Rights Statement Identifier

Definition	**linkingRightsStatementIdentifier** – This section holds the identifier of a copyrights statement that may be stored outside of Rosetta.
Rosetta Mandatory	No
Source	User
Repeatable	Yes (no limits)
Level	IE, Representation or File
METS section	rightsMD

Collection

Definition	**collection** – This section holds the information of the collection(s) that the IE is associated with. There could be multiple records pointing to multiple collections/sub-collections. The collection METS will have one record that holds the identifiers of the collection and the parent collection (if exists).
Rosetta Mandatory	No
Source	User
Repeatable	Yes for IE (no limits), no in case of collection METS
Level	IE, Collection
METS section	techMD

The full list of fields in each section is specified in Appendix B – DNX Data Dictionary.