

# Metadata

## Content

[Descriptive metadata](#)  
[Identifying metadata](#)  
[Structural metadata](#)  
[Technical metadata](#)  
[Logging of preservation actions](#)  
[Administrative metadata](#)

## Further information

[Submission information packages \(SIP\)](#)

[Archival information packages \(AIPs\)](#)

[Dissemination information packages \(DIPs\)](#)

[Preservation Management](#)

[Rosetta product documentation:](#)  
Rosetta DNX Profile in *Rosetta AIP Data Model*

[Rosetta METS Profile](#)

## Descriptive metadata

Descriptive metadata is recorded in the digital long-term archiving system with the aim of uniquely describing and identifying the objects. The descriptive metadata is intended to ensure the long-term assignment of the content of the object and is created by the relevant TIB specialist teams, delivered directly to the long-term archiving team by the data producers or collected by the long-term archiving team from various data sources. Descriptive metadata in the DC section of ie.xml (see [Specifications for archival information packages \(AIPs\)](#)) must be available as Dublin Core. This metadata is indexed. Various metadata standards (MARC, Dublin Core, MODS, EAD, NISO, MIX and others) can be integrated into the source MD section of ie.xml.

There are currently several methods for recording descriptive metadata:

- enrichment with metadata from the union catalog K10plus
- the collection of Dublin Core metadata supplied via the OAI interface from the institutional repository of Leibniz Universität Hannover
- the capture of supplied Dublin Core metadata in the dc section of ie.xml
- the collection of supplied metadata from source systems as source metadata in ie.xml

Additional catalogue systems may be connected as required.

## Enrichment with metadata from the union catalogue K10plus

Librarians collect metadata on the objects according to the [RDA cataloguing standard](#). Older catalogue records are available based on the RAK-WB standard.

CMS enrichment is conducted during the transition from operational to permanent archival storage. CMS enrichment involves querying metadata via the SRU interface of the Gemeinsamer Verbundkatalog and mapping the output to Dublin Core. Mapping governs the assignment of [PICA+](#) (only in german) fields to the relevant Dublin Core qualified elements, as well as the scope, structure and content of the descriptive metadata.

The metadata are written to a separate catalogue.xml and given an identifier; the identifier is written to the ie.xml. The metadata from the catalogue.xml are indexed.

## Mapping table from PICA+ to Dublin Core

Dublin Core	Pica+	Remark	Mandatory
<b>Title</b>	036C/00	Collective title of the multi-part monograph and the subcategorisations (in master form)	Yes, in combination with 021A
<b>isPartOf</b>	036C/00	Collective title of the multi-part monograph and the subcategorisations (in master form)	No
<b>title</b>	021A	Main title, other title information, information on responsibilities	Yes
<b>alternative</b>	046B	Specification of parallel titles that are not on the title page	No
<b>Alternative</b>	021F*	Parallel titles	No
<b>Alternative</b>	046C*	Deviating titles	No
<b>Creator</b>	028A	Person/family as first creator (formerly: first author)	No
<b>Creator</b>	028B/..	Second author and additional authors	No
<b>creator or contributor</b>	028C /00*	Person/family as additional creators, other contributing persons and families	No
<b>Creator</b>	029A	Body / first originator	No
<b>Contributor</b>	028M	Creator from superordinate C set	No
<b>Contributor</b>	028G-028L	Other person, dedicatee (old prints), censor (old prints), artistic contributor (old prints), other non-involved persons or persons named in the title (old prints)	No

<b>creator or contributor</b>	029F /00*	Secondary body, other bodies involved	No
<b>Contributor</b>	030F*	Congress	No
<b>Publisher</b>	033A*	Publication details (place of publication and publisher)	No
<b>Publisher</b>	037C*	Note in university publications	No
<b>Issued</b>	011@	Publication date	No
<b>language</b>	010@	Language codes	No
<b>identifier</b>	005A*	ISSN	No
<b>identifier</b>	004U*	Persistent identifier: URN	No
<b>identifier</b>	004V	Persistent identifier: DOI	No
<b>identifier</b>	004R*	Persistent identifier: Handle	No
<b>identifier</b>	004A*	ISBN	No
<b>identifier</b>	007F*	Report number	No
<b>identifier</b>	007G	ID number given by first cataloguing institution (EKI)	Yes
<b>identifier</b>	003@	PICA production number (PPN)	No
<b>isPartOf</b>	036E*	Monographic series	No
<b>isPartOf</b>	036F*	Monographic series (link)	No
<b>isPartOf</b>	039B*	Link to larger entity (in the case of articles)	No
<b>Bibliographic Citation</b>	031A	Differentiating information about the source	No
<b>description</b>	032@	Edition statement	No
<b>description</b>	032B	Reprint note	No

## Metadata delivered by data producers or harvested by a platform

For metadata supplied by the data producer and harvested by platforms, the Long-Term Preservation team has defined minimum sets for different forms of publication in Dublin Core. Metadata that is not available as Dublin Core can be included in the archive package as source metadata.

### Monographs

Content	captured in	mandatory
title	dc:title	✓
author names (repeatable)	dc:creator / dc:contributor	✓
ISBN	dc:identifier xsi:type="dcterms:ISBN"	✗
DOI	dc:identifier xsi:type="dcterms:URI"	✗
other unique identifiers (repeatable)	dc:identifier	✗
language	dc:language	✗
publication year	dcterms:issued	✗
abstract	dcterms:abstract	✗
publisher	dc:publisher	✗

## Journal articles

Content	captured in	mandatory
article title	dc:title	✓
author names (repeatable)	dc:creator / dc:contributor	✓
journal title; <i>volume</i> , <i>issue</i> , publication year	dcterms:isPartOf	✓✗✗✗
<i>DOI</i>	dc:identifier xsi:type="dcterms:URI"	✗
<i>ISSN</i>	dc:identifier xsi:type="dcterms:ISSN"	✗
<i>language</i>	dc:language	✗
<i>publication year</i>	dcterms:issued	✗
<i>abstract</i>	dcterms:abstract	✗

## Identifying metadata

### Identifiers used

#### Internal-system identifiers at the object level

Rosetta creates and allocates various internal-system identifiers.

- Identifier for objects: system internal identifier generated by Rosetta to identify IEs, representations, files and packets during deposit and SIP processing.
- Event type identifier: Rosetta-defined ID for an event category (see [Event](#)).
- Identifier for processes: ID assigned by Rosetta for executed processes, for example a Preservation Action (see [Administrative metadata](#) and [Logging of preservation actions](#)).
- Rights identifier: the ID of a policy, for example, a configured usage right (see ), a retention policy, or a delivery license.
- Identifier for agents: the ID of an agent in the sense of PREMIS, for example, a producer, a plug-in, a connected system, or a user.

The internal-system identifiers are unique and permanent within the system.

If new policies or processes are defined by a user, the system assigns a new unique ID. Additional identifiers are recorded in the metadata.

### Catalogue metadata

Another optional external identifier in the ie.xml is the catalogue identifier from the Gemeinsamer Verbundkatalog (Union Catalogue, GVK). By means of the SRU interface to the catalogue system, configured in Rosetta, the catalogue identifier is used to enrich the object with descriptive metadata.

The catalogue metadata of each individual object are deposited in a dedicated XML file, which is linked to the IE via metadata identifiers (mld)

Identifiers are allocated PREMIS-compliant for objects, agents, events and rights. The following table lists several examples of identifiers.

#### Examples of identifiers based on the PREMIS model

Object	Example
<b>SIP ID</b>	539308
<b>IE ID</b>	IE2980431
<b>REP ID</b>	REP2980432
<b>File ID</b>	FL2980433

<b>Identifier for the catalogue system</b>	GBV881139254
<b>mld</b>	1032839
<b>Versioning</b>	V9-IE1024027.xml
<b>Agent</b>	
<b>Producer ID</b>	40030044
<b>Producer agent ID</b>	2122740
<b>Plug-in ID</b>	58638365
<b>Catalogue system</b>	TIB
<b>User ID</b>	2122740
<b>Event</b>	
<b>Material flow ID</b>	641084
<b>Deposit ID</b>	548243
<b>Event ID</b>	62
<b>Process ID</b>	50532321
<b>Rights</b>	
<b>Boilerplate ID</b>	TIB_OA_mit_CC
<b>Access right policy ID</b>	16728
<b>Retention policy ID</b>	NO_RETENTION

## External identifiers

External identifiers can be recorded in Dublin Core format, such as a DOI, a handle or a URN.

## Allocation of identifiers

Internal-system identifiers are automatically allocated by the system as unique identifiers. The identifiers are given different additions, depending on the object type.

## Structural metadata

Structural metadata are stored in the ie.xml as DNX and METS elements.

TIB stores 1-n representations per IE, each consisting of 1-n files. Representations are described using the DNX element "Preservation type". Each ie.xml contains the IDs of all associated representations and files. In the file group, files are assigned to a file ID via their path, and each file ID is also assigned to a representation ID. In the StructMap, the files per representation are arranged in a logical sequence that can be transferred to a viewer.

## Structural metadata assignment to METS and DNX elements

Metadatum	Element and metadata standard	Value
<b>Representations</b>		
<b>Original files</b>	Preservation type (DNX)	MASTER

<b>Modified copy of original files before ingest</b>	Preservation type (DNX)	PRE-INGEST_MODIFIED_MASTER
<b>Modified copy of original files after ingest</b>	Preservation type (DNX)	MODIFIED_MASTER
<b>Access copy</b>	Preservation type (DNX)	DERIVATIVE_COPY
<b>Relationships</b>		
<b>Belonging of files to a representation</b>	fileGrp (METS)	REP ID, File ID, storage path to the file
<b>Coherence of files within a representation</b>	structMap (METS)	Representation ID, label structure, file ID
<b>Restoration of authentic data structure</b>		
<b>Original file name</b>	fileOriginalName (DNX)	Original file name
<b>Original file path</b>	fileOriginalPath (DNX)	Original file path

The relationships between files within a representation are recorded in the “structMap” METS element. In addition, the original file name and path of every file are recorded in the metadata, documenting which directory structure a file was stored in during deposit.

## Technical metadata

Technical metadata are captured in Rosetta as DNX metadata. DNX was specified by the software manufacturer Ex Libris and is based on PREMIS, but extends the standard by further elements. [DNX documentation](#) is publicly available. Updating of DNX is managed and monitored by the Rosetta user community.

The [PREMIS](#) standard defines a number of “basic concepts” as technical metadata in the semantic units ObjectCharacteristics, SignificantProperties, OriginalName and Storage. The relevant concepts of the unit are provided in the table below. In this case, the PREMIS concept is mapped to the DNX element, as well as information about at which point the concept can be allocated values and whether TIB has implemented the recording.

### Technical metadata mapping from PREMIS to DNX

<b>PREMIS semantic unit / component from</b>	<b>DNX element</b>	<b>Method of recording</b>	<b>Used by TIB</b>
<b>ObjectCharacteristics</b>			
<b>compositionLevel</b>	compositionLevel	Pre-ingest	No
<b>fixity</b>			
<b>messageDigestAlgorithm</b>	fileFixity. fixityType	See K10	Yes
<b>messageDigest</b>	fileFixity. agent	See K10	Yes
<b>messageDigestOriginator</b>	fileFixity. fixityValue	See K10	Yes
<b>size</b>	generalFileCharacteristics. fileSizeBytes	Determined automatically during ingest	Yes
<b>format</b>			
<b>formatDesignation</b>			

<b>formatName</b>	fileFormat. formatName	Automatically during ingest	Yes
<b>formatVersion</b>	fileFormat. formatVersion	Automatically during ingest	Yes
<b>formatRegistry</b>			
<b>formatRegistryName</b>	fileFormat. formatRegistry	Automatically during ingest	Yes
<b>formatRegistryKey</b>	fileFormat. formatRegistryId	Automatically during ingest	Yes
<b>formatRegistryRole</b>	fileFormat. formatRegistryRole	Automatically during ingest	Yes
<b>formatNote</b>	fileFormat. formatNote	Manually by the technical analyst during ingest involving manual allocation to format	Yes
<b>creatingApplication</b>			
<b>Last name</b>	creatingAp plication. creatingAp plicationNa me	As part of the pre- ingest process, manually via the web editor or automatically as part of a preservation plan	No  TIB does not use this semantic concept to capture the creatingApplication, but records the values – provided they can be recorded by the technical metadata extractor – under significant properties as part of the technical metadata
<b>Version</b>	creatingAp plication. creatingAp plicationVe rsion	See above	See above
<b>dateCreatedByApplication</b>	creatingAp plication. dateCreate dByApplica tion	See above	See above
<b>creatingApplicationExtension</b>	creatingAp plication. creatingAp plicationEx tension	See above	See above
<b>inhibitors</b>			
<b>inhibitorType</b>	inhibitors. inhibitorTy pe	As part of the pre- ingest process or manually via the web editor	Yes
<b>inhibitorTarget</b>	inhibitors. inhibitorTar get	See above	See above
<b>inhibitorKey</b>	inhibitors. inhibitorKey	See above	See above
<b>significantProperties</b>			
<b>significantPropertiesType</b>	significant Properties Type	Metadata extraction in the validation stack	Yes
<b>significantPropertiesValue</b>	significant Properties Value	See above	Yes
<b>significantPropertiesExtension</b>	significant Properties Exten	See above	Yes

<b>originalName</b>	fileOriginal Name	Automatically during ingest	Yes
	fileOriginal Path	Automatically during ingest	Yes
<b>storage</b>			
<b>contentLocation</b>			
<b>contentLocationType</b>	fileLocation Type	Automatically during ingest (system – loading stage)	Yes
<b>contentLocationValue</b>	fileLocation	Is not used by Rosetta at present.	

## Logging of preservation actions

### Defined events

Modifications to AIPs are recorded at the IE level as DNX metadata. The DNX schema was specified by the software manufacturer Ex Libris and is based on PREMIS, but extends the standard by additional elements. [DNX documentation](#) is publicly available. Updating of DNX is managed and monitored by the Rosetta user community.

Several examples of defined events are described in the table below. The complete list of defined events is documented in the [Rosetta Configuration Guide](#).

### Examples of events

Event ID	Description
23	Started Validation Stack Stage
24	Virus check performed on file
25	Format Identification performed on
27	Fixity check performed on file
147	Arranger Decline IE
164	Object viewing is denied due to Access Rights restrictions
165	Technical Metadata extraction performed on file
166	Completed Validation Stack Stage
167	Metadata enrichment (CMS fetching)
217	Failed MD Validation Stage
339	Preservation plan has been created
372	Manually Set Format Library ID on File
380	Representation has been added
381	Risk identification performed on file
397	METS Validation Failed

A user with the role of “Administrator” can define which events from the list should be logged.

## Logging of event metadata

The system automatically records the defined event metadata. Event metadata are written to the ie.xml for every defined event.

## Administrative metadata

### Defined administrative metadata

Administrative metadata are captured as DNX metadata at different levels in Rosetta. DNX was specified by the software manufacturer Ex Libris and is based on PREMIS, but extends the standard by further elements. [DNX documentation](#) is publicly available.

At the IE level, the standardised name of the applicable licence agreement is recorded as the Dublin Core element dctersm:license. The applicable licence text is deposited in Rosetta as a “boilerplate”; the text contains information about which actions may be performed on the object.

TIB understands administrative metadata to mean:

- Metadata that document the provenance of objects
- Legal metadata
- Metadata recorded for the purpose of organising objects

### Provenance information

Provenance information	DNX element
Acquisition team responsible	producer
	producerId
	userIdAppld
	defaultLanguage
	authorativeName
	firstName
	lastName
	middleName
	address1
	address2
	address3
	address4
	zip
	emailAddress
	telephone1

### Legal metadata

Legal metadata	Element
----------------	---------



<b>Access rights</b>	accessRightsPolicy (DNX)
	policyId (DNX)
	policyDescription (DNX)
<b>Title of the transfer agreement as concluded between TIB and the data producer or the long-term archiving team and the transferring TIB team, or standardized name of the applicable license text.</b>	Dcterms: license (Dublin Core)
<b>Access right to the document as granted by the data producer/rights holder /copyright holder</b>	dcterms: accessRights
<b>Legal basis for long-term archiving</b>	dc:rights
<b>Right of use in trigger case</b>	dc:rights
<b>Authorized users in trigger case</b>	dcterms: accessRights
<b>Rights holder</b>	dcterms: rightsHolder

## Organisational metadata

Organisational metadata	DNX element
<b>General object characteristics (at the IE representation and file level, respectively)</b>	<b>objectCharacteristics</b>
	ObjectType
	parentID
	groupID
	creationDate
	createdBy
	modificationDate
	modifiedBy
	owner
<b>IE characteristics</b>	<b>generalIECharacteristics</b>
	submissionReason
	status
	statusDate
<i>Identification of object type</i>	IEEntityType
<i>Identifier for the collection and production process</i>	UserDefinedFieldA
<i>Marking for non-valid or password-protected objects in the context of Preservation as a Service</i>	UserDefinedFieldB
<i>Marking for images from defective media devices</i>	UserDefinedFieldC

<b>Preservation level</b>	<b>preservationLevel</b>
	preservationLevelValue
<b>Representation characteristics</b>	<b>generalRepCharacteristics</b>
	label
	preservationType
	usageType