Metadaten

Inhalt

- Deskriptive Metadaten
 - Anreicherung mit Metadaten aus dem Verbundkatalog K10plus
 - Von Datenproduzenten abgelieferte oder von einer Plattform geharvestete Metadaten
- Identifizierende Metadaten
 - Eingesetzte Identifier
 - Vergabe von Identifier
- Strukturelle Metadaten
 - Strukturelle Metadaten - Zuordnung zu METS- und DNX-Elementen
- Technische Metadaten
 - Technische Metadaten - Mapping von PREMIS auf DNX
- Protokollierung von Erhaltungsmaßnahmen
 - Definierte Events
 - Protokollierung von Eventmetadaten
- Administrative Metadaten
 - Definierte administrative Metadaten

Weiterführende Informationen

- [Ablieferungspakete \(SIP\)](#)
- [Archivinformationspakete \(AIP\)](#)
- [Nutzungspakete \(DIP\)](#)
- [Erhaltungsplanung \(Preservation Management\)](#)
- [Rosetta Produktdokumentation: Rosetta DNX Profile im Rosetta AIP Data Model](#)
- [Rosetta METS Profile](#)

Deskriptive Metadaten

Im digitalen Langzeitarchivierungssystem werden beschreibende Metadaten mit dem Ziel erfasst, die Objekte eindeutig beschreiben und identifizieren zu können. Die beschreibenden Metadaten sollen langfristig die inhaltliche Zuordnung des Objekts gewährleisten und werden von den entsprechenden Fachteams der TIB erstellt, von den Datenproduzenten direkt an das Team Langzeitarchivierung abgeliefert oder vom Team Langzeitarchivierung von verschiedenen Datenquellen abgeholt. Beschreibende Metadaten in der DC-Sektion der ie.xml (siehe [Spezifikation für Archivinformationspakete \(AIP\)](#)) müssen als Dublin Core vorhanden sein. Diese Metadaten werden indexiert. In die Source MD-Sektion der ie.xml können verschiedene Metadatenstandards (MARC, Dublin Core, MODS, EAD, NISO, MIX und weitere) eingebunden werden.

Es gibt aktuell mehrere Verfahren für die Erfassung von deskriptiven Metadaten:

- die Anreicherung mit Metadaten aus dem Verbundkatalog K10plus
- die Erfassung von über die OAI-Schnittstelle mitgelieferten Dublin Core Metadaten vom Institutionellen Repository der Leibniz Universität Hannover
- die Erfassung von mitgelieferten Dublin Core Metadaten in der dc-Section der ie.xml
- die Erfassung von mitgelieferten Metadaten aus Quellsystemen als Source-Metadaten in der ie.xml

Weitere Katalogsysteme können bei Bedarf angebunden werden.

Anreicherung mit Metadaten aus dem Verbundkatalog K10plus

Bibliothekarinnen und Bibliothekare erfassen Metadaten zu den Objekten nach dem [Katalogisierungsstandard RDA](#). Ältere Katalogisate liegen nach dem Standard [RAK-WB](#) vor.

Beim Übergang vom operativen Speicher in den permanenten Archivspeicher wird das CMS-Enrichment ausgeführt, das über die SRU-Schnittstelle des Verbundkataloges K10plus Metadaten abfragt und den Output auf Dublin Core mappt. Ein Mapping regelt die Zuordnung der [PICA+](#)-Felder zu den entsprechenden Dublin-Core-Qualified-Elementen sowie den Umfang, die Struktur und den Inhalt der beschreibenden Metadaten.

Die Metadaten werden in eine separate Katalog.xml geschrieben, mit einem Identifier benannt und der Identifier in die ie.xml geschrieben. Die Metadaten aus der Katalog.xml werden indexiert.

Mappingtabelle von PICA+ auf Dublin Core

Dublin Core	Pica+	Bemerkung	Pflicht
Title	036C/00	Gesamtittel der mehrteiligen Monografie und der Untergliederungen (in Vorlageform)	ja, Kombination mit 021A
isPartOf	036C/00	Gesamtittel der mehrteiligen Monografie und der Untergliederungen (in Vorlageform)	nein
title	021A	Haupttitel, Zusätze, Verantwortlichkeitsangabe	ja
alternative	046B	Angabe von Paralleltiteln, die nicht auf der Haupttitelseite stehen	nein
Alternative	021F*	Paralleltitel	nein
Alternative	046C*	abweichende Titel	nein
Creator	028A	Person/Familie als 1. geistiger Schöpfer (früher: 1. Verfasser)	nein
Creator	028B/..	2. und weitere Verfasser	nein
creator oder contributor	028C /00*	Person/Familie als weitere geistige Schöpfer, sonstige beteiligte Personen und Familien	nein
Creator	029A	Körperschaft/1. Urheber	nein
Contributor	028M	Verfasser aus übergeordnetem C-Satz	nein
Contributor	028G-028L	sonstige Person, Widmungsempfänger (alte Drucke), Zensor (alte Drucke), künstlerischer Beiträger (alte Drucke), sonstige Nichtbeteiligte bzw. im Titel genannte Personen (alte Drucke)	nein

creator oder contributor	029F /00*	Sekundärkörperschaft, sonstige beteiligte Körperschaft	nein
Contributor	030F*	Kongress	nein
Publisher	033A*	Veröffentlichungsangabe (Erscheinungsort und Verlag)	nein
Publisher	037C*	Hochschulschriftenvermerk	nein
Issued	011@	Erscheinungsdatum	nein
language	010@	Sprachcodes	nein
identifier	005A*	ISSN	nein
identifier	004U*	Persistent Identifier: URN	nein
identifier	004V	Persistent Identifier: DOI	nein
identifier	004R*	Persistent Identifier: Handle	nein
identifier	004A*	ISBN	nein
identifier	007F*	Reportnummer	nein
identifier	007G	Identnummer der erstkatalogisierenden Institution (EKI)	ja
identifier	003@	PICA-Produktionsnummer (PPN)	nein
isPartOf	036E*	monografische Reihe	nein
isPartOf	036F*	monografische Reihe (Verknüpfung)	nein
isPartOf	039B*	Verknüpfung zur größeren Einheit (bei Aufsätzen)	nein
Bibliographic Citation	031A	differenzierende Angaben zur Quelle	nein
description	032@	Ausgabebezeichnung	nein
description	032B	Reprintvermerk	nein

Von Datenproduzenten abgelieferte oder von einer Plattform geharvestete Metadaten

Für vom Datenproduzenten mitgelieferte und von Plattformen abgeholte Metadaten hat das Team Langzeitarchivierung Minimalsets für unterschiedliche Publikationsformen in Dublin Core definiert. Metadaten, die nicht als Dublin Core vorliegen, können als Source Metadaten in das Archivpaket aufgenommen werden.

Monografien

Inhalt	erfasst in	Pflicht
Titel	dc:title	✓
Autorennamen (wiederholbar)	dc:creator / dc:contributor	✓
ISBN	dc:identifier xsi:type="dcterms:ISBN"	✗
DOI	dc:identifier xsi:type="dcterms:URI"	✗
weitere eindeutige Identifier	dc:identifier	✗
Sprache	dc:language	✗
Erscheinungsjahr	dcterms:issued	✗
Beschreibung/Abstract	dcterms:abstract	✗

Verlag/Herausgeber	dc:publisher	✘
--------------------	--------------	---

Zeitschriftenartikel

Inhalt	erfasst in	Pflicht
Artikeltitel	dc:title	✓
Autorennamen (wiederholbar)	dc:creator / dc:contributor	✓
Zeitschriftentitel ; <i>Jahrgang/Volume</i> , <i>Issue</i> , Jahr	dcterms:isPartOf	✓✘✘✘
<i>DOI</i>	dc:identifier xsi:type="dcterms:URI"	✘
<i>ISSN</i>	dc:identifier xsi:type="dcterms: ISSN"	✘
<i>Sprache</i>	dc:language	✘
<i>Erscheinungsdatum/-jahr</i>	dcterms:issued	✘
<i>Beschreibung/Abstract</i>	dcterms:abstract	✘

Identifizierende Metadaten

Eingesetzte Identifier

Systeminterne Identifier auf Objektebene

Rosetta erzeugt und vergibt verschiedene systeminterne Identifier.

- Identifier für Objekte: von Rosetta erzeugte systeminterne Identifier zur Identifizierung von IEs, Repräsentationen, Files und Paketen während des Deposits und der SIP-Prozessierung.
- Identifier für Eventtyp: von Rosetta fest vorgegebene ID für ein Eventkategorie (siehe [Event](#)).
- Identifier für Prozesse: von Rosetta vergebene ID für ausgeführte Prozesse, zum Beispiel eine Preservation Action (siehe [Administrative Metadaten](#) und [Protokollierung von Erhaltungsmaßnahmen](#))
- Identifier für Rechte: die ID einer Policy, zum Beispiel eines konfigurierten Nutzungsrechts (siehe), einer Aufbewahrungsfrist (Retention Policy) oder einer Ablieferungslizenz.
- Identifier für Agents: die ID eines Agenten im Sinne von PREMIS, zum Beispiel eines Producers, eines Plug-ins , eines angebundenen Systems oder eines Users.

Die systeminternen Identifier sind innerhalb des Systems eindeutig und dauerhaft.

Werden neue Policies oder Prozesse von einem User definiert, vergibt das System dafür eine eindeutige ID. In den Metadaten werden weitere Identifier erfasst.

Katalogmetadaten

Ein weiterer optionaler externer Identifier in der ie.xml ist der Katalog-Identifier aus dem K10plus. Mittels in Rosetta konfigurierter SRU-Schnittstelle zum Katalogsystem wird der Katalog-Identifier zur Anreicherung des Objekts mit deskriptiven Metadaten genutzt.

Die Katalogmetadaten eines jeden Objekts werden in einer eigenen XML-Datei abgelegt, welche mittels Metadaten-Identifier (mld) mit der IE verknüpft ist.

Identifier werden PREMIS-konform für Objects, Agents, Events und Rights vergeben. Die folgende Tabelle listet exemplarisch Beispiele für Identifier auf.

Beispiele für Identifier anhand des PREMIS-Modells

Object	Beispiel
SIP-ID	539308
IE-ID	IE2980431

REP-ID	REP2980432
File-ID	FL2980433
Identifizier zum Katalogsystem	GBV881139254
mId	1032839
Versionierung	V9-IE1024027.xml
Agent	
Producer-ID	40030044
Producer Agent ID	2122740
Plug-in-ID	58638365
Katalogsystem	TIB
User-ID	2122740
Event	
Material Flow-ID	641084
Deposit-ID	548243
Event-ID	62
Prozess-ID	50532321
Rights	
Boilerplate-ID	TIB_OA_mit_CC
Access Right Policy-ID	16728
Retention Policy ID	NO_RETENTION

Externe Identifizier

Externe Identifizier können im Dublin-Core-Format aufgenommen werden, zum Beispiel ein DOI, ein Handle oder eine URN.

Vergabe von Identifizieren

Systeminterne Identifizier werden vom System als Unique Identifier automatisch vergeben. Je nach Objekttyp erhalten die Identifizier verschiedene Zusätze.

Strukturelle Metadaten

Strukturelle Metadaten werden in der ie.xml als DNX- und METS-Elemente erfasst.

Die TIB erfasst pro IE 1-n Repräsentationen, die aus 1-n Dateien bestehen. Repräsentationen werden mit dem DNX-Element „Preservation Type“ beschrieben. Jede ie.xml beinhaltet die IDs aller zugehörigen Repräsentationen und Files. In der File Group werden Files via ihres Pfades einer File-ID zugeordnet, zusätzlich wird jede File-ID einer Repräsentations-ID zugeordnet. In der StructMap werden die Files pro Repräsentation in eine logische Reihenfolge gebracht, die an einen Viewer übergeben werden kann.

Strukturelle Metadaten - Zuordnung zu METS- und DNX-Elementen

Metadatum	Element und Metadatenstandard	Wert
-----------	-------------------------------	------

Repräsentationen		
Originaldateien	Preservation Type (DNX)	MASTER
Modifizierte Kopie von Originaldateien vor dem Ingest	Preservation Type (DNX)	PRE-INGEST_MODIFIED_MASTER
Modifizierte Kopie von Originaldateien nach dem Ingest	Preservation Type (DNX)	MODIFIED_MASTER
Nutzungskopie	Preservation Type (DNX)	DERIVATIVE_COPY
Beziehungen		
Zugehörigkeit von Dateien zu einer Repräsentation	fileGrp (METS)	REP-ID, File-ID, Speicherpfad zur Datei
Zusammenhang von Dateien innerhalb einer Repräsentation	structMap (METS)	Repräsentations-ID, Labelstruktur, File-ID
Wiederherstellung der authentischen Datenstruktur		
Originaldateiname	fileOriginalName (DNX)	Originaldateiname
Originaler Dateipfad	fileOriginalPath (DNX)	originaler Dateipfad

Die Beziehungen der Dateien innerhalb einer Repräsentation werden in dem METS-Element „structMap“ erfasst. Zusätzlich wird von jeder Datei der Originaldateiname und -pfad in den Metadaten erfasst und so dokumentiert, in welcher Verzeichnisstruktur eine Datei beim Deposit abgelegt war.

Technische Metadaten

Technische Metadaten werden in Rosetta als DNX-Metadaten erfasst. DNX wurde vom Softwarehersteller ExLibris spezifiziert und basiert auf PREMIS, erweitert den Standard jedoch um weitere Elemente. Die [Dokumentation von DNX](#) ist öffentlich einsehbar. Die Fortschreibung von DNX wird von der Rosetta User Community gesteuert und überwacht.

Der PREMIS-Standard selbst definiert eine Reihe von „Basiskonzepten“ als technische Metadaten in den Semantischen Einheiten ObjectCharacteristics, SignificantProperties, OriginalName und Storage. Die entsprechenden Konzepte der Einheit sind in nachstehender Tabelle aufgeführt. Es erfolgt hier ein Mapping des PREMIS-Konzepts auf das DNX-Element sowie die Angabe, an welcher Stelle das Konzept mit Values versehen werden kann und der Hinweis, ob die Erfassung aktuell von der TIB umgesetzt ist.

Technische Metadaten - Mapping von PREMIS auf DNX

PREMIS Semantic Unit / Component aus	DNX-Element	Erfassungsmethode	Von TIB genutzt
ObjectCharacteristics			
compositionLevel	compositionLevel	Pre-Ingest	nein
fixity			
messageDigestAlgorithm	fileFixity.fixityType	siehe K10	ja
messageDigest	fileFixity.agent	siehe K10	ja
messageDigestOriginator	fileFixity.fixityValue	siehe K10	ja
size	generalFileCharacteristics.fileSizeBytes	automatisch im Ingest ermittelt	ja

format			
formatDesignation			
formatName	fileFormat. formatName	automatisch im Ingest	ja
formatVersion	fileFormat. formatVersion	automatisch im Ingest	ja
formatRegistry			
formatRegistryName	fileFormat. formatRegistry	automatisch im Ingest	ja
formatRegistryKey	fileFormat. formatRegistryId	automatisch im Ingest	ja
formatRegistryRole	fileFormat. formatRegistryRole	automatisch im Ingest	ja
formatNote	fileFormat. formatNote	manuell von Technischem Analyst im Ingest bei manueller Zuweisung zu Format	ja
creatingApplication			
Name	creatingAp plication. creatingAp plicationNa me	als Teil vom Pre-Ingest, manuell über Web- Editor oder automatisch als Teil eines Preservation Plans	nein Die TIB nutzt zur Erfassung der creatingApplication nicht dieses Semantic Concept, sondern erfasst die Values – insofern vom technischen Metadata Extractor erfassbar – als Teil der technischen Metadaten unter Significant Properties
Version	creatingAp plication. creatingAp plicationVe rsion	siehe oben	siehe oben
dateCreatedByApplication	creatingAp plication. dateCreate dByApplica tion	siehe oben	siehe oben
creatingApplicationExtension	creatingAp plication. creatingAp plicationExt ension	siehe oben	siehe oben
inhibitors			
inhibitorType	inhibitors. inhibitorType	Als Teil vom Pre-Ingest oder manuell via Web- Editor	Ja
inhibitorTarget	inhibitors. inhibitorTar get	siehe oben	siehe oben
inhibitorKey	inhibitors. inhibitorKey	siehe oben	siehe oben
significantProperties			
significantPropertiesType	significantP ropertiesTy pe	Metadatenextraktion im Validation Stack	Ja
significantPropertiesValue	significantP ropertiesVa lue	siehe oben	Ja

significantPropertiesExtension	significantPropertiesExtension	siehe oben	Ja
originalName	fileOriginalName	automatisch im Ingest	ja
	fileOriginalPath	Automatisch im Ingest	ja
storage			
contentLocation			
contentLocationType	fileLocationType	automatisch im Ingest (System – Loading stage)	ja
contentLocationValue	fileLocation	Wird aktuell von Rosetta nicht genutzt.	

Protokollierung von Erhaltungsmaßnahmen

Definierte Events

Änderungen an AIPs werden auf IE-Ebene als DNX-Metadaten erfasst. Das DNX-Schema wurde vom Softwarehersteller ExLibris spezifiziert und basiert auf PREMIS, erweitert den Standard jedoch um weitere Elemente. Die [Dokumentation von DNX](#) ist öffentlich einsehbar. Die Fortschreibung von DNX wird von der Rosetta User Community gesteuert und überwacht.

Einige Beispiele für definierte Events sind in der folgenden Tabelle beschrieben. Die vollständige Liste mit definierten Events ist im [Rosetta Configuration Guide](#) dokumentiert.

Beispiel-Events

Event-ID	Beschreibung
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

Ein User mit der Rolle „Administrator“ kann definieren, welche Events aus der Liste protokolliert werden sollen.

Protokollierung von Eventmetadaten

Die definierten Eventmetadaten werden automatisiert vom System erfasst. Zu jedem definierten Event werden Eventmetadaten in die ie.xml geschrieben.

Administrative Metadaten

Definierte administrative Metadaten

Administrative Metadaten werden in Rosetta auf verschiedenen Ebenen als DNX-Metadaten erfasst. DNX wurde vom Softwarehersteller ExLibris spezifiziert und basiert auf PREMIS, erweitert den Standard jedoch um weitere Elemente. Die [Dokumentation von DNX](#) ist öffentlich einsehbar.

Auf IE-Ebene wird der standardisierte Name des geltenden Lizenztextes als Dublin-Core-Element `dc:license` erfasst. Der geltende Lizenztext wird als sogenanntes Boilerplate in Rosetta hinterlegt und beinhaltet die Informationen, welche Aktionen am Objekt vorgenommen werden dürfen.

Die TIB versteht unter administrativen Metadaten:

- Metadaten, die die Provenienz der Objekte dokumentieren
- rechtliche Metadaten
- Metadaten, die zum Zweck der Organisation der Objekte erfasst werden

Provenienzinformation

Provenienzinformation	DNX-Element
Zuständiges Erwerbsteam	producer
	producerId
	userIdAppld
	defaultLanguage
	authorativeName
	firstName
	lastName
	middleName
	address1
	address2
	address3
	address4
	zip
	emailAddress
	telephone1

Rechtliche Metadaten

Rechtliche Metadaten	Element
----------------------	---------

Recht der Verbreitung durch die TIB	accessRightsPolicy (DNX)
	policyId (DNX)
	policyDescription (DNX)
Titel der Übereignahmevereinbarung, wie zwischen TIB und Datenproduzent:innen bzw. Team Langzeitarchivierung und abgebendem TIB-Team abgeschlossen oder standardisierter Name des geltenden Lizenztextes	Dcterms: license (Dublin Core)
Zugriffsrecht auf das Dokument, so wie vom Datenproduzent:in/Rechteinhaber:in /Urheber:in erteilt	dcterms: accessRights
Rechtsgrundlage für die Langzeitarchivierung	dc:rights
Nutzungsrecht im Triggerfall	dc:rights
Berechtigte Nutzer:in im Triggerfall	dcterms: accessRights
Rechteinhaber:in	dcterms: rightsHolder

Organisatorische Metadaten

Organisatorische Metadaten	DNX-Element
allgemeine Objekteigenschaften (jeweils auf IE- Repräsentations- und Dateiebene)	objectCharacteristics
	ObjectType
	parentID
	groupID
	creationDate
	createdBy
	modificationDate
	modifiedBy
	owner
IE-Eigenschaften	generalIECharacteristics
	submissionReason
	status
	statusDate
<i>Kennzeichnung der Objektart</i>	IEEntityType
<i>Kennzeichnung für die Sammlung und den Produktionsweg</i>	UserDefinedFieldA
<i>Kennzeichnung für nicht-valide oder passwortgeschützte Objekte im Dienstleistungskontext</i>	UserDefinedFieldB

<i>Kennzeichnung für Images von defekten Datenträgern</i>	UserDefinedFieldC
Preservation Level	preservationLevel
	preservationLevelValue
Repräsentationseigenschaften	generalRepCharacteristics
	label
	preservationType
	usageType