

LMER

Langzeitarchivierungsmetadaten für elektronische Ressourcen

Version: 1.2

Stand: 7. April 2005

URN: urn:nbn:de:1111-2005041102

Redaktion: Tobias Steinke

© 2005

Die Deutsche Bibliothek (Deutsche Bücherei Leipzig, Deutsche Bibliothek Frankfurt am Main, Deutsches Musikarchiv Berlin)

Inhaltsverzeichnis

Inhaltsverzeichnis	3
Einführung	4
I. Motivation	4
II. Konzepte	5
Kern von technischen Metadaten	5
Referenzen zur Beschreibung der Systemumgebung.....	5
Modularisierung zur Einbindung in andere Strukturen	6
III. Release Notes	7
Referenz	8
I. Übersicht über die Elemente	8
II. Ausführliche Referenzbeschreibung	9
1. Objekt (ImerObject)	9
2. Prozess (ImerProcess)	12
3. Datei (ImerFile)	14
4. Metadaten Modifizierung (ImerModification).....	18
Ansprechpartner	19

Einführung

I. Motivation

Elektronische Dokumente stellen für Bibliotheken neue Medien dar. Die Inhalte sind jedoch in der Regel vergleichbar mit herkömmlichen Büchern und Schriften. Somit greifen für die inhaltsbezogenen Beschreibungen die herkömmlichen bibliografischen Daten, wie sie in Deutschland derzeit über Datenformate wie MAB2 und MARC21 ausgetauscht werden. Unabhängig davon sind für elektronische Dokumente aber weitere Metadaten nötig, die insbesondere für die Langzeitarchivierung entscheidend sind. Während sich die bibliografischen Daten in einem Katalogsystem befinden, welches für thematisches Recherchieren des Nutzers gedacht ist, sollten die technischen Metadaten direkt bei den elektronischen Dokumenten in einem Depotsystem gespeichert werden, wo sie für die Anwendung von Methoden zur Langzeitarchivierung (Emulation und Migration) genutzt werden können.

Dieses Dokument will solche technischen Metadaten speziell zur Langzeitarchivierung definieren. Dabei wird versucht, dem Stand der Forschung zum Thema Langzeitarchivierung Rechnung zu tragen, zugleich aber auch praxisrelevant und implementierungsnah zu sein.

Allgemein unbestritten ist, dass moderne Metadaten in XML beschrieben werden, da sich dieses Syntax-Format weltweit als Standard durchgesetzt hat. Was die konkrete Ausgestaltung angeht, so gibt es verschiedene theoretische Vorarbeiten (vorrangig sei hier das europäische Projekt NEDLIB und die OCLC/RLG-Arbeitsgruppe „Preservation Metadata Framework“ genannt). Was die praktische Ausarbeitung angeht, so haben vor allem zwei Ansätze Relevanz: PREMIS¹, ein sehr neuer Ansatz (Anfang 2005) für einen internationalen Standard, sowie das „Preservation Metadata: Metadata Implementation Schema“² der National Library of New Zealand vom Juli 2003.

Auf diesen Grundlagen werden im Folgenden die Langzeitarchivierungsmetadaten für elektronische Ressourcen (LMER) vorgestellt, wobei insbesondere das Datenmodell aus Neuseeland als Basis fungiert. Ziel ist ein universelles Format für technische Metadaten, syntaktisch beschrieben durch ein XML-Schema. LMER ist also kein allgemeines Datenmodell für Langzeitarchivierungsmetadaten, sondern vielmehr für den konkreten Einsatz in Austauschformaten konzipiert. Wie diese Metadaten in einem Depotsystem gespeichert und verwaltet werden ist damit nicht vorgegeben. LMER kann sowohl eigenständig als Austauschformat für technische Metadaten genutzt werden, als auch als Teil anderer XML-Strukturen.

¹ <http://www.oclc.org/research/projects/pmwg/>

² <http://www.natlib.govt.nz/en/whatsnew/4initiatives.html#meta>

II. Konzepte

Kern von technischen Metadaten

Jede Dateiart, z. B. Bild, Video, Tonaufnahme, Text, erfordert spezielle technische Metadaten. Welche technischen Kenngrößen dabei jeweils anfallen, ändert sich mit der fortschreitenden Entwicklung. Somit ist der Versuch problematisch, ein umfassendes Metadaten-set zu entwickeln, das alle erdenklichen Metadaten für alle Dateiarten beinhaltet. LMER geht deshalb einen anderen Weg als das Metadata Implementation Schema der National Library of New Zealand. Es wurde versucht, einen Kern von Metadaten zu finden, der alle Dateiarten betrifft. Um spezifische technische Metadaten aufzunehmen, wurde in Abschnitt „Datei“ von LMER ein Feld vorgesehen (xmlData), welches beliebige XML-Metadaten in einem anderen Schema aufnehmen kann. Dieser modulare Ansatz, welcher durch das Namespace-Konzept von XML ermöglicht wird, macht LMER zukunftssicher und vielseitiger. Im praktischen Einsatz müssen allerdings Absprachen getroffen werden, welches Metadaten-schema bzw. ob überhaupt ein solches für eine bestimmte Dateiart zum Einsatz kommt.

Referenzen zur Beschreibung der Systemumgebung

Eine wichtige Information im Sinne der Langzeitarchivierung ist die Beschreibung des genauen Formats und der nötigen Systemumgebung, um eine Wiedergabe in der ursprünglich gedachten Form zu ermöglichen. Das umfasst zahlreiche Aspekte, sowohl der Hardware als auch der Software. Da diese Informationen wiederum voneinander abhängen – z. B. kann ein PDF-Dokument einer bestimmten Version in allen dafür vorgesehenen Acrobat Readern dargestellt werden, die wiederum für bestimmte Betriebssysteme verfügbar sind, welche wiederum bestimmte Anforderungen an die Hardware haben – ist es wenig sinnvoll, diese einzeln in jeder Datei abzuspeichern. Vielmehr sollte eine (inter-/national) zentrale Datenbank gepflegt werden, die solche Abhängigkeiten speichert, und auf die in den Metadaten der einzelnen Dateien Verweise zeigen.

Somit reicht es, eine eindeutige Kennung aus einer solchen File Format Registry mitzugeben. Steht als Format etwa „PDF 1.2“ in LMER, dann wäre das eine Referenz in eine solche zentrale Datenbank, die wiederum bei Bedarf Informationen liefern kann, welche Programme dieses Format erstellen und anzeigen können, auf welchen Betriebssystemen diese Programme jeweils laufen und welche Hardwarevoraussetzungen nötig sind. Auch die anderen Ebenen dieser Datenbank könnten referenzgeeignete Bezeichnungen liefern. Steht in LMER als Erzeugerprogramm z. B. „OpenOffice.org 1.1“, dann könnte die Datenbank mögliche Betriebssystem- und Hardwareanforderungen dazu nennen.

Derzeit gibt es verschiedene internationale Bestrebungen für solche Datenbanken von Dateiformaten. Schon funktionstüchtig, aber bisher ohne geeignete eindeu-

tige Referenzen, ist PRONOM³. Die Nationalbibliothek der Niederlande benutzt im Rahmen ihres Systems DIAS eine eigene File Format Registry als Teil des Preservation Managers.

Modularisierung zur Einbindung in andere Strukturen

Metadaten zur Langzeitarchivierung werden in der Regel im Kontext mit anderen Metadaten benutzt. Um diese verschiedenen Metadaten zu einem Objekt in einem Austauschformat zu bündeln, benötigt es weitere Vereinbarungen. Eine solche ist der Metadata Encoding & Transmission Standard (METS)⁴. METS dient zur Beschreibung der Struktur eines Objekts und kann zudem Abschnitte mit Metadaten inhaltlicher als auch technischer Natur enthalten. Zur Integration in METS und ähnliche XML-Metadaten austauschformate teilt sich LMER in mehrere XML-Schemas auf. Somit kann LMER eigenständig als Austauschformat für technische Metadaten genutzt werden oder als Teil eines umfassenderen Austauschformates.

Als Beispiel für die Integration in METS wird hier ein mögliches Vorgehen skizziert:

- METS enthält insbesondere in der File Section bestimmte Attribute und Felder, die ebenfalls in LMER als Entsprechungen auftauchen. Zugunsten der Kompatibilität mit anderen METS-Daten werden deshalb alle LMER-Felder (auch Pflichtfelder) weggelassen, die bereits durch entsprechende METS-Felder beschrieben sind.
- Daten aus dem LMER-Abschnitt Objekt finden sich im METS-Abschnitt Administrative Metadata und dort im Unterabschnitt Technical Metadata. Dieser Unterabschnitt wird in der betroffenen File Group-Beschreibung der METS-File Section referenziert.
- Daten aus dem LMER-Abschnitt Datei finden sich im METS-Abschnitt Administrative Metadata und dort im Unterabschnitt Technical Metadata. Dieser Unterabschnitt wird in der betroffenen File-Beschreibung der METS-File Section referenziert.
- Daten aus dem LMER-Abschnitt Process finden sich im METS-Abschnitt Administrative Metadata und dort im Unterabschnitt Digital Provenance Metadata. Dieser Unterabschnitt wird in der betroffenen File Group-Beschreibung (bei Bezug auf das Objekt) oder der betroffenen File-Beschreibung (bei Bezug auf eine Datei) der METS-File Section referenziert.
- Der LMER-Abschnitt Modifikation wird weggelassen.
- Referenzen in LMER-Feldern, die sich auf das LMER-Feld Datei-ID beziehen (Einstiegsdatei, Abhängigkeit), benennen stattdessen die ID der entsprechenden File-Beschreibung der METS-File Section.

³ <http://www.records.pro.gov.uk/pronom/>

⁴ <http://www.loc.gov/standards/mets/>

III. Release Notes

Version 1.0: Erste öffentliche Version

Version 1.1: Interne Version

- Für das Konzept der Modularisierung wird das XML-Schema in mehrere für die einzelnen Abschnitte aufgeteilt.
- Einführung des Felder Abhängigkeit (linkedTo) im Abschnitt Datei und Status im Abschnitt Objekt.

Version 1.2: Zweite öffentliche Version

- Der Abschnitt Prozess kann sowohl Unterabschnitt von Objekt als auch von Datei sein.
- Neue Felder in Abschnitt Prozess: oldMetadataRecordCreator, oldObjectIdentifier, oldObjectVersion
- Die Felder Format und Transferformat bekommen das Attribut FORMATREGISTRYNAME zur Spezifikation der referenzierten File Format Registry.
- Die Felder fileChecksumType und transferChecksumType werden ersetzt durch entsprechende Attribute bei fileChecksum und transferChecksum.
- Alle Felder, die eine Datumsangabe enthalten, sind nun im Format ISO 8601 anzugeben (XML-Datentyp xsd:dateTime).
- Folgende Felder sind jetzt wiederholbar: format, transferFormat, fileChecksum, transferChecksum, groupIdentifier, formatInfos

Referenz

I. Übersicht über die Elemente

LMER gliedert sich in folgende Abschnitte, die sich am Metadata Implementation Schema von Neuseeland orientieren:

Objekt: Darunter fallen Metadaten, die sich gemeinsam auf alle Teildateien des Dokuments beziehen. Darin befindet sich auch die URN als Persistent Identifier, die den eindeutigen Bezug zu den bibliografischen Metadaten herstellt.

Prozess: In diesen Metadaten werden sämtliche technischen Veränderungen an einem Objekt oder einzelnen Dateien eines Objektes protokolliert. Der Abschnitt Prozess ist entweder ein Unterabschnitt von Objekt oder von Datei und kann dort jeweils mehrfach vorkommen. Für jede Änderung gibt es einen eigenen Prozess-Abschnitt.

Datei: Für jede Datei, die zum Objekt gehört, finden sich hier eigene Metadaten zur Beschreibung der Charakteristiken. Dabei setzen sich die Metadaten aus allgemeinen Feldern zusammen, die für jeden Dateityp gleich sind, und aus spezifischen (z. B. die Framerate für Videos), welche als spezialisierte Metadaten aus anderen Schemata im Feld xmlData eingebunden werden. Der Abschnitt Datei ist ein Unterabschnitt von Objekt und kann dort mehrfach vorkommen. Für jede zum Objekt gehörende Datei gibt es einen eigenen Datei-Abschnitt.

Metadaten Modifizierung: In diesen Metadaten werden sämtliche Veränderungen an den LMER-Metadaten selbst protokolliert. Es werden keine Änderungen in Metadaten berücksichtigt, die sich nicht in der gleichen Datei befinden.

II. Ausführliche Referenzbeschreibung

Unterstrichene Felder sind obligatorisch, wenn LMER nicht als Teil einer anderen Struktur wie METS eingesetzt wird. Der Abschnitt Objekt muss immer vorhanden sein. Ist „Anzahl der Dateien“ größer als Null, dann muss es exakt genauso viele Datei-Abschnitte geben. Alle anderen Abschnitte sind optional. Obligatorische Felder in diesen Abschnitten sind nur dann nötig, wenn der Abschnitt vorhanden ist.

„NLNZ“ im Feld „Benutzte Standards“ verweist auf gleiche Felder im Metadata Implementation Schema der National Library of New Zealand. In Klammern dahinter findet sich der entsprechende Feldname und ggf. Hinweise, ob es Unterschiede zu der dortigen Definition gibt.

1. Objekt (*ImerObject*)

Objekt-ID (objectIdentifier)

Definition: Eindeutige Kennung des zugehörigen Objekts.

Benutzte Standards: NLNZ (Object Identifier, aber anderer Datentyp)

Datentyp: String

Beispiel: obj123

Bemerkung: Anders als der Persistent Identifier braucht die Eindeutigkeit dieser Kennung nicht über die zu archivierende Institution hinauszugehen. Hinweis, aus welchem Kontext diese ID kommt, gibt das Feld metadata-RecordCreator.

Name des Objekts (name)

Definition: Eine für Menschen lesbare Bezeichnung des Objekts

Benutzte Standards: NLNZ (Name of Object)

Datentyp: String

Beispiele: Der fröhliche Jäger, Ag. 06/1995

Bemerkung: Muss nicht einmalig sein, da maschinell nicht als Identifikation benutzt

Persistent Identifier (persistentIdentifier)

Definition: International eindeutiger Bezeichner für das Objekt

Benutzte Standards: NLNZ (Persistent Identifier)

Datentyp: String

Beispiel: urn:nbn:de:gbv:104-opus-291

Bemerkung: Hier ist vor allem an die Verwendung von URNs (siehe <http://www.persistent-identifier.de/>) gedacht, aber andere Konzepte zu Persistent Identifiern (z. B. DOI) können in diesem Feld auch ihren Platz finden. Diese Kennung kann z. B. benutzt werden, um eine Verbindung zu extern gespeicherten bibliografischen Metadaten zu definieren.

Transfer-URL (transferURL)

Definition: URL zu einer transportfähigen Form des Objekts (zusammengefasst in einer Datei, z. B. gepackt als ZIP-Datei)

Benutzte Standards: URL

Datentyp: String

Beispiel: ftp://ftp.bib-test.de/ddb/doc1234567890.zip

Bemerkung: Die Adresse muss nicht permanent gültig sein

Transferformat (transferFormat)

Definition: Bezeichnung des Formats der Datei zu Feld Transfer-URL als exakte Referenz in eine externe Datenbank

Benutzte Standards: analog PREMIS (formatRegistryKey)

Datentyp: String

Beispiel: PDF 1.2

Bemerkung: Dies ist kein Freitext, sondern eine möglichst präzise Formatbeschreibung, die aus einem wohldefinierten Wertebereich stammt. Welcher Wertebereich referenziert wird, gibt ein Attribut (REGISTRYNAME) an. Dieses Attribut kann ein formaler Name oder auch eine URI sein. Je nach Aussagekraft von Transferformat (z. B. als URI) kann es unnötig sein. Das Feld ist wiederholbar.

Transfer-MIME-Format (transferMimeType)

Definition: MIME-Format der Datei zu Feld Transfer-URL

Benutzte Standards: MIME, siehe

<http://www.iana.org/assignments/media-types/index.html>

Datentyp: String

Beispiel: application/zip

Bemerkung: -

Transferchecksumme (transferChecksum)

Definition: Checksumme der Datei bei Transfer-URL

Benutzte Standards: CRC32, MD5, SHA-1, etc.

Datentyp: String

Beispiel: 304ac95579c21f3498dfdf7117d3845220d34f

Bemerkung: Ein Attribut (CHECKSUMTYPE) gibt die verwendete Checksummenart an. Für die derzeit meist benutzten Methoden sollten exakt folgende Schreibweisen im Attribut benutzt werden: CRC32, MD5, SHA-1. Dieses Feld ist wiederholbar.

Gruppen-ID (groupIdentifier)

Definition: Eine Kennung, um zusammengehörige Objekte zu markieren

Benutzte Standards: NLNZ (Group Identifier, aber anderer Datentyp), PREMIS (preservationLevel)

Datentyp: String

Beispiel: dbbszz11089

Bemerkung: Für die Zusammengehörigkeit ist vorwiegend an technische Aspekte gedacht, z. B. Bilder, die exakt unter gleichen Bedingungen im Zuge einer bestimmten Aktion generiert wurden oder Objekte, die nur für bestimmte Erhaltungsmaßnahmen vorgesehen sind (z. B. nur Emulation,

keine Migration). Dieses Feld ist wiederholbar, ein Objekt kann also mehreren Gruppen angehören.

Objektversion (objectVersion)

Definition: Wenn es verschiedene Archivobjekte desselben Originalobjekts gibt (als Folge einer internen Migration), dann findet sich hier ein schneller Indikator für die Version.

Benutzte Standards: -

Datentyp: String

Beispiel: 2

Bemerkung: Hiermit ist keine inhaltliche Version gemeint, sondern nur die technisch andere Darstellung des gleichen Objekts. Wird bei einer Migration die Ausgangsdatei behalten (hängt vom Depotsystem ab), dann hat das neue Objekt (Ergebnis der Migration) einen eigenen Datensatz an LMER-Daten. In diesen findet sich in den Prozess-Abschnitten die Beschreibung der Änderung, nicht in den LMER-Daten des noch im Depotsystem vorhandenen, aber unveränderten Ausgangsobjekts.

Entstehungsdatum des Archivobjekts (masterCreationDate)

Definition: Zeitpunkt, an dem aus dem Originalobjekt das Archivobjekt hergestellt wurde

Benutzte Standards: NLNZ (Preservation Master Creation Date)

Datentyp: Date/Time (ISO 8601)

Beispiel: 2003-11-24T16:25:00

Bemerkung: Für einen Webschnitt wäre das der Zeitpunkt des Harvest-Vorgangs

Entstehungsdatum der Metadaten (metadataCreationDate)

Definition: Datum (und Zeit) der Entstehung der technischen Metadaten

Benutzte Standards: NLNZ (Date of Metadata Record Creation)

Datentyp: Date/Time (ISO 8601)

Beispiel: 2003-11-24T16:28:00

Bemerkung: -

Ersteller der Metadaten (metadataRecordCreator)

Definition: Person oder System oder Organisation, durch den/das/die die technischen Metadaten erstellt wurden

Benutzte Standards: NLNZ (Metadata Record Creator, aber geänderte Semantik)

Datentyp: String

Beispiel: LMER Analyzer 1.2

Bemerkung: Bei einem System sollte dieses genau spezifiziert sein (mit Versionsnummer). Dieses Feld benennt den Kontext, in dem objectIdentifier definiert ist.

Einstiegsdatei (startFile)

Definition: ID einer Datei, die bei mehreren Dateien den Einstieg bietet

Benutzte Standards: Ähnlich NLNZ (Target Indicator, aber geänderte Semantik und anderer Datentyp)

Datentyp: String

Beispiel: file0123

Bemerkung: Die zugehörige ID muss als fileIdentifier in einem Datei-Abschnitt definiert sein, hier steht kein Dateiname.

Anzahl der Dateien (numberOfFiles)

Definition: Gesamtzahl der Dateien, die zu diesem Objekt gehören

Benutzte Standards: -

Datentyp: Positive Integer

Beispiel: 5

Bemerkung: Die Anzahl muss der Anzahl der Datei-Abschnitte entsprechen, Ordner werden selbst nicht als Dateien gezählt

Status (status)

Definition: Textfeld für einen objektbezogenen Status

Benutzte Standards: -

Datentyp: String

Beispiel: deleted

Bemerkung: Dieses Feld spielt vor allem bei der Verwendung als Austauschformat zwischen verschiedenen Archivsystemen eine Rolle.

Kommentar (comments)

Definition: Freitextfeld für Ergänzungen

Benutzte Standards: NLNZ (Comments)

Datentyp: String

Beispiel: Objekt wurde als Test erstellt

Bemerkung: Hier sollten nur Bemerkungen stehen, die sich auf das ganze Objekt beziehen

2. Prozess (ImerProcess)

Ersteller der alten Metadaten (oldMetadataRecordCreator)

Definition: Person oder System oder Organisation, durch den/das/die die technischen Metadaten der vorangegangenen Migration erstellt wurden

Benutzte Standards: -

Datentyp: String

Beispiel: LMER Analyzer 1.2

Bemerkung: Bei einem System sollte dieses genau spezifiziert sein (mit Versionsnummer). Dieses Feld benennt den Kontext, in dem oldObjectIdentifier definiert ist.

Alte Objekt-ID (oldObjectIdentifier)

Definition: Eindeutige Kennung des Objekts der vorangegangenen Migration.

Benutzte Standards: -

Datentyp: String

Beispiel: obj122

Bemerkung: Anders als der Persistent Identifier, der für jede Migration identisch ist, ändert sich die Objekt-ID für jede Konvertierung. Hinweis, aus welchem Kontext diese ID kommt, gibt das Feld oldMetadataRecordCreator.

Alte Objektversion (oldObjectVersion)

Definition: Versionsnummer der vorangegangenen Migration.

Benutzte Standards: -

Datentyp: String

Beispiel: 1

Bemerkung: Diese Versionsnummer hilft beim Nachvollziehen der Migrationshistorie.

Veränderungsgrund (purpose)

Definition: Beschreibung der Ursache für die vorgenommene Änderung

Benutzte Standards: NLNZ (Purpose)

Datentyp: String

Beispiel: Format xyz ist obsolet und wird nicht mehr unterstützt

Bemerkung: Dies ist ein Freitext, der möglichst nachvollziehbar ausgefüllt werden sollte. Evtl. könnte dabei eine Art Liste von Vorgaben hinzugezogen werden.

Änderer (processCreator)

Definition: Person oder System, welche/s die Änderung vorgenommen hat

Benutzte Standards: NLNZ (Person / Agency Performing Process, aber geänderte Semantik)

Datentyp: String

Beispiel: Preservation Toolbox V2.0

Bemerkung: -

Erteiler der Veränderungsgenehmigung (permission)

Definition: Person, entweder ein Vorgesetzter des Änderers oder ein Verantwortlicher für das System, welches als Änderer eingetragen wurde

Benutzte Standards: NLNZ (Permission)

Datentyp: String

Beispiel: Zimmermann, Robert

Bemerkung: Hier sollte immer eine Person beschrieben werden, kein weiteres System

Datum der Erteilung der Veränderungsgenehmigung (permissionDate)

Definition: Datum (und Zeit) der Veränderungsverantwortlichkeit

Benutzte Standards: NLNZ (Permission Date)

Datentyp: Date/Time (ISO 8601)

Beispiel: 2003-11-24T17:30:00

Bemerkung: -

Veränderungsschritte (steps)

Definition: Beschreibung des genauen Vorgehens bei der Veränderung

Benutzte Standards: NLNZ (Steps)

Datentyp: String

Beispiel: 1. Analyse des Formates xyz, 2. Verlustfreie Umwandlung in das Format zyx, 3. Überprüfung des entstandenen Formats zyx

Bemerkung: Handelt es sich um ein standardisiertes Verfahren, kann auch dessen Benennung mit eindeutiger Referenz (z. B. URN) als Beschreibung reichen.

Ergebnisse (result)

Definition: Beschreibung des durch die Änderung erreichten Status

Benutzte Standards: NLNZ (Result)

Datentyp: String

Beispiel: Alle zum Objekt gehörigen Dateien im Format xyz wurden verlustfrei in das Format zyx umgewandelt.

Bemerkung: -

Änderungsdatum (completionDate)

Definition: Datum (und Zeit) der Erreichung der genannten Ergebnisse

Benutzte Standards: NLNZ (Completion Date/Time)

Datentyp: Date/Time (ISO 8601)

Beispiel: 2003-11-24T18:00:00

Bemerkung: -

Kommentar (comments)

Definition: Ergänzende Kommentare

Benutzte Standards: NLNZ (Comments)

Datentyp: String

Beispiel: Da die erfolgte Migration verlustfrei verlief, wird keine Kopie des Originals behalten.

Bemerkung: -

3. Datei (ImerFile)

Datei-ID (fileIdentifier)

Definition: Eindeutige Kennung der zugehörigen Datei

Benutzte Standards: NLNZ (File Identifier)

Datentyp: String

Beispiel: file0123

Bemerkung: Diese Kennung dient auch als Verknüpfung zur Strukturbeschreibung (z. B. wie in METS)

Verzeichnis (path)

Definition: Ordnerstruktur, in dem die Datei stehen muss, damit sie nutzbar ist

Benutzte Standards: NLNZ (File Path)

Datentyp: String

Beispiel: /docs/pdf/

Bemerkung: Fehlt dieses Feld, so entspricht das einem Verzeichnis „/“ (also auf der ersten Ebene des Objekts). Bei mehreren Dateien, die sich nicht alle auf derselben, ersten Ebene befinden, ist dieses Feld obligatorisch.

Dateiname (name)

Definition: Vollständiger Name der Datei (ohne Verzeichnis)

Benutzte Standards: NLNZ (Filename & Extension)

Datentyp: String

Beispiel: dissertation.pdf

Bemerkung: -

Dateigröße (size)

Definition: Größe der Datei in Bytes

Benutzte Standards: NLNZ (File Size)

Datentyp: Positive Integer

Beispiel: 529123

Bemerkung: -

Dateidatum (fileDateTime)

Definition: Datum (und Zeit) der Entstehung der Datei

Benutzte Standards: NLNZ (File Date/Time)

Datentyp: Date/Time (ISO 8601)

Beispiel: 2002-12-24T12:00:00

Bemerkung: -

Dateichecksumme (fileChecksum)

Definition: Checksumme der Datei

Benutzte Standards: CRC32, MD5, SHA-1, etc.

Datentyp: String

Beispiel: 304ac95579c21f3498dfdf7117d3845220d34f

Bemerkung: Ein Attribut (CHECKSUMTYPE) gibt die verwendete Checksummenart an. Für die derzeit meist benutzten Methoden sollten exakt folgende Schreibweisen im Attribut benutzt werden: CRC32, MD5, SHA-1. Dieses Feld ist wiederholbar.

MIME-Format (mimeType)

Definition: Beschreibung des Formats der Datei nach dem MIME-Standard

Benutzte Standards: NLNZ (MIME Type)

Datentyp: String

Beispiel: application/pdf

Bemerkung: -

Format (format)

Definition: Bezeichnung des Formats der Datei als exakte Referenz in eine externe Datenbank

Benutzte Standards: analog PREMIS (formatRegistryKey)

Datentyp: String

Beispiel: PDF 1.2

Bemerkung: Dies ist kein Freitext, sondern eine möglichst präzise Formatbeschreibung, die aus einem wohldefinierten Wertebereich stammt. Welcher Wertebereich referenziert wird, gibt ein Attribut (REGISTRYNAME) an. Dieses Attribut kann ein formaler Name oder auch eine URI sein. Je nach Aussagekraft von Format (z. B. als URI) kann es unnötig sein. Das Feld ist wiederholbar.

Ergänzende Formatinformationen (formatInfos)

Definition: Zusätzliche Informationen zur exakten Formatbestimmung

Benutzte Standards: -

Datentyp: String

Beispiel: Linearized PDF, ISO PDF/X-1, ISO PDF/X-1a

Bemerkung: Der Wert im Feld Format sollte zur eindeutigen Bestimmung eigentlich ausreichen. Trotzdem können wichtige Informationen zu bestimmten Formaten vorhanden sein (weil sie z. B. von einem Tool wie JHOVE geliefert wurden), die in diesem Feld bewahrt werden können. . Das Feld ist wiederholbar.

Erstellungsprogramm (creatorApplication)

Definition: Bezeichnung des Erstellungsprogramms der Datei, idealerweise als exakte Referenz in eine externe Datenbank

Benutzte Standards: PREMIS (creatingApplicationName, creatingApplicationVersion)

Datentyp: String

Beispiel: OpenOffice.org 1.1

Bemerkung: Dies ist im Idealfall kein Freitext, sondern eine möglichst präzise Programmbeschreibung, die aus einem wohldefinierten Wertebereich stammt. In jedem Fall sollte der Programmname nur zusammen mit der Versionsnummer benutzt werden.

Darstellungsprogramm (viewerApplication)

Definition: Bezeichnung eines Darstellungsprogramms der Datei, idealerweise als exakte Referenz in eine externe Datenbank

Benutzte Standards: analog PREMIS (creatingApplicationName, creatingApplicationVersion)

Datentyp: String

Beispiel: Adobe Reader 6.0

Bemerkung: Dies ist im Idealfall kein Freitext, sondern eine möglichst präzise Programmbeschreibung, die aus einem wohldefinierten Wertebereich stammt. In jedem Fall sollte der Programmname nur zusammen mit der Versionsnummer benutzt werden. Dieses Feld hat nicht den Anspruch der Vollständigkeit, sondern soll bewusst ein Programm nennen, von dem die korrekte Anzeige der Datei bekannt ist.

Abhängigkeit (linkedTo)

Definition: Datei-ID einer anderen Datei desselben Objektes, bei deren Änderung auch diese Datei geändert werden muss

Benutzte Standards: NLNZ (File Identifier)

Datentyp: String

Beispiel: file0124

Bemerkung: Bei Migrationen soll dieses Feld helfen, nötige Abhängigkeiten zu erkennen. Z. B. hängt eine HTML-Datei von allen verlinkten Dateien ab, da sich bei deren Migration in der Regel die Dateinamensendung ändert und somit auch die zugehörigen Links. Bei den Abhängigkeiten sind nur solche zu Dateien desselben Objekts gemeint, keine zu anderen Objekten. Das Feld ist wiederholbar.

Kommentar (comments)

Definition: Ergänzende Kommentare

Benutzte Standards: -

Datentyp: String

Beispiel: Diese Datei ließ sich auch korrekt im Acrobat Reader 5.0 darstellen.

Bemerkung: -

Kategorieerkennung (category)

Definition: Bezeichnung des Typs der Datei und zugleich Indikator, für die weiteren Metadaten

Benutzte Standards: DINI-Empfehlung für die Bezeichnung des Dokumenttyps bei der OAI-Set-Bildung (<http://www.dini.de/dokumente.php>)

Datentyp: String

Beispiel: Text

Bemerkung: Der Wertebereich ist {text, notes, image, audio, video, multimedia, data, binary}

Es folgen spezifische technische Metadaten für die jeweilige Kategorie. Für Bilder bietet sich zum Beispiel MIX⁵ an, für Texte textMD⁶. Diese Metadaten sind keine Elemente von LMER und gehören daher zu einem eigenen XML-Namespaces.

Spezifische technische Metadaten (xmlData)

Definition: Block mit vom Format abhängigen technischen Metadaten

Benutzte Standards: XML-Metadaten

Datentyp: XML-Daten

Beispiel: <mix> ... </mix>

Bemerkung: Über ein verpflichtendes Attribut MDTYPE wird der exakte Name der verwendeten Metadaten bezeichnet. Es können mehrere Blöcke mit spezifischen technischen Metadaten vorhanden sein, das Feld ist also wiederholbar.

⁵ <http://www.loc.gov/standards/mix/>

⁶ <http://dlib.nyu.edu/METS/textmd.htm>

4. Metadaten Modifizierung (lmerModification)

Änderer (modifier)

Definition: Person oder System, welche/s die Änderung vorgenommen hat

Benutzte Standards: NLNZ (Metadata Record Modifier)

Datentyp: String

Beispiel: Preservation Toolbox V2.0

Bemerkung: -

Änderungsdatum (dateTime)

Definition: Datum (und Zeit) der Änderung

Benutzte Standards: NLNZ (Date/Time)

Datentyp: Date/Time (ISO 8601)

Beispiel: 2003-11-24T18:00:00

Bemerkung: Es werden alle Änderungen an den Metadaten (mit Ausnahme des eigenen Abschnitts), auch den nicht zu LMER-gehörigen, vermerkt. Allerdings müssen die Metadaten im selben XML-Dokument vorliegen.

Modifiziertes Feld (fieldModified)

Definition: Exakte Beschreibung des geänderten XML-Felds

Benutzte Standards: NLNZ (Field Modified, aber anderer Datentyp), XPath, siehe <http://www.w3.org/TR/xpath>

Datentyp: String

Beispiel: /mets/amdSec/lmer/object/process[@id = P1234]/steps

Bemerkung: Die Darstellung in XPath funktioniert nur in einer XML-Struktur. Bei einer Speicherung innerhalb einer Datenbank, müssen die Metadatenfelder mit internen Kennungen referenziert werden.

Geänderte Daten (data)

Definition: Wert des geänderten Felds vor der Änderung

Benutzte Standards: NLNZ (Data Modified)

Datentyp: String

Beispiel: Migration

Bemerkung: Ggf. muss der Wert in String konvertiert werden

Kommentar (comments)

Definition: Ergänzende Kommentare

Benutzte Standards: -

Datentyp: String

Beispiel: Beschreibung war vorher zu ungenau

Bemerkung: Hier kann auch ein Grund für die Änderung vermerkt werden.

Ansprechpartner

Tobias Steinke, Die Deutsche Bibliothek, Deutsche Bibliothek Frankfurt am Main
E-Mail: steinke@dbf.ddb.de