Specifications for submission information packages (SIP)

Content

Further information

Package structures

Transfer package specifications

TIB uses several levels of transfer packages, which are described here. The graphic Package structure provides a general overview of the transfer packages used:

- Transfer information packages
- Pre-ingest SIPs
- Post-ingest SIPs

Specifications for transfer information packages are relevant for the delivery of objects.

The graphic Transforming transfer information packages to SIPs and AIPs describes how transfer information packages are transformed to pre-ingest SIPs, post-ingest SIPs and AIPs.

Transfer information packages

Different transfer information structures are used for different scenarios:

1. SIPs with a simple structure and one representation
   SIPs with a simple structure can be submitted as ZIP files or as folders.
2. METS deposit
   a. for objects with multiple representations or complex data structures
   b. Objects with multiple representations or complex data structures and externally created METS file
3. Connection to a repository via an OAI interface
4. CSV-Deposit
   for objects with complex relationships that are to be mapped as a collection or with metadata from source systems

The transfer information packages are described in the form of normalised tables. The table below explains the structure of the normalised tables.

Specification parameter	Implementation
Naming convention	Naming convention according to which the package must be named
Package structure	Structure in which the package must be available
Content data	Description of the minimum and maximum number of files expected
Permissible file formats	Description of the permissible file formats, where relevant
Representations	The number and type of representations allowed
Quality of data	Describes whether only valid and well-formed files are accepted
Metadata	Describes whether or not the object must be indexed in the Gemeinsamer Verbundkatalog (Union Catalogue, GVK)
Identifier	Identifier that uniquely identifies the object and links it to descriptive metadata, such as a PPN (identification number), an EKI (identification number given by the first cataloguing institution) or a handle
Legal metadata	Describes whether the object belongs to a collection in which several licence texts, licence text versions and access rights can be allocated. If this is the case, this must be mapped in a superordinate directory structure.

Objects with a simple structure and a representation

Example: the German Research Report team

With this transfer information structure, there must be exactly one file that belongs to the MASTER representation.

Specification parameter	Implementation
Naming convention	The file is named using the PPN.
Package structure	One PDF file per SIP
Content data	Exactly one PDF file
Permissible file formats	Exactly one PDF file
Representations	MASTER
Quality of data	Non-valid and non-well-formed files are also accepted.
Metadata	The object must be indexed in the catalogue.
Identifier	An identifier that refers to a catalogue record is expected to be found in the name of the PDF file.
Legal metadata	Objects are sorted by licence agreement and access rights before ingest.

Objects with multiple representations or complex data structures

In the transfer structure for complex objects, different representations may be ingested with 1-n files each.

<table>
<tr><th>Specification parameter</th><th>Implementation</th></tr>
<tr><td>Naming convention</td><td>The package is named at the top directory level using the EKI (identification number given by the first cataloguing institution).</td></tr>
<tr><td>Package structure</td><td>For each SIP, there is a directory named using the EKI. It contains a directory for each representation; the directory is named based on the name vocabulary defined in the archive. The content data are available in the representation folders.
<pre>
IDENTIFIER
|--MASTER (mandatory)
| |--File1
| |--File n
| |-- Folder 0-n
| |--File 0-m
|--MODIFIED_MASTER (optional)
| |--File1
| |--File n
| |-- Folder 0-n
| |--File 0-m
|--DERIVATIVE_COPY (optional)
| |--File1
| |--File n
| |-- Folder 0-n
| |--File 0-m
</pre></td></tr>
<tr><td>Content data</td><td>At least one file per representation</td></tr>
<tr><td>Permissible file formats</td><td>No limitation</td></tr>
<tr><td>Representations</td><td>MASTER, MODIFIED_MASTER, DERIVATIVE_COPY</td></tr>
</table>

Quality of data	Non-valid and non-well-formed files are also accepted.
Metadata	The object must be indexed in the catalogue.
Identifier	EKI
Legal metadata	The acquisition team uses a superordinate directory structure to assign the objects to the collection groups, publications types, applicable licence texts (in different versions, where applicable) and access rights.

Objects with multiple representations or complex data structures and externally created METS file

In the transfer structure for complex objects with externally created METS file, the path identifier/content/streams may contain different representations with 1-n files each. A METS file which validates against the Rosetta xsd (https://developers.exlibrisgroup.com/rosetta/integrations/mets-dnx/) must be handed over.

<table>
<tr><th>Specification parameter</th><th>Implementation</th></tr>
<tr><td><b>Naming convention</b></td><td>The package is named at the top directory level using an unique identifier.</td></tr>
<tr><td><b>Package structure</b></td><td>

For each SIP, there is a directory named with an unique identifier. It contains one dc.xml with Dublin Core metadata and a directory named content. content directory contains a METS file ie1.xml and a directory streams, which contains the actual data. If multiple representations are handed over, the relevant files must be allocated to the corresponding representation in the METS file.

Other representation names than MASTER, PRE-INGEST_MODIFIED_MASTER and DERIVATIVE_COPY must be coordinated with TIB’s digital preservation team.

```
IDENTIFIER
|--dc.xml
|--content
|        |--ie1.xml
|        |--streams
|                |--File1
|                |--File n
|                |--Folder 0-n
|                        |--File 0-m
```

</td></tr>
<tr><td><b>Content data</b></td><td>At least one file per representation.</td></tr>
<tr><td><b>Permissible file formats</b></td><td>No limitation</td></tr>
<tr><td><b>Representations</b></td><td>MASTER (mandatory), PRE-INGEST_MODIFIED_MASTER (optional), DERIVATIVE_COPY (optional), further representations according to prior agreement</td></tr>
<tr><td><b>Quality of data</b></td><td>Non-valid and non-well-formed files are also accepted.</td></tr>
<tr><td><b>Metadata</b></td><td>The object has not be indexed in the catalogue. A dc.xml must be provided.</td></tr>
<tr><td><b>Identifier</b></td><td>Unique identifier</td></tr>
</table>

Legal metadata	Legal metadata are captured as follows: 1) The access right to the object as assigned by the depositing institution shall be documented as dcterms:accessRights (e.g. private/public or another controlled vocabulary) 2) The depositing institution‘s right to preserve the object shall be documented as dc:rights 3) The submission agreement as concluded between TIB‘s team digital preservation and the depositing institution shall be documented as dcterms: license.

Connection to a repository via an OAI interface using the example of Leibniz Universität Hannover Institutional Repository

With this transfer information package, records are ingested via the OAI interface of Leibniz Universität Hannover Institutional Repository; the records must contain the metadata, at least a title and an identifier, and 1-n files. All objects within a record belong to the MASTER representation.

Specification parameter	**Implementation**
Naming convention	The original file name of the file is kept. No specification check is performed.
Package structure	At least one file per SIP
Content data	At least one file At least one metadata format must include direct links to the files and supplements belonging to the record.
Permissible file formats	No limitation
Representations	MASTER
Quality of data	Non-valid and non-well-formed files are also accepted.
Metadata	The repository must contain at least the object’s title and identifier metadata. There may also be additional metadata.
Identifier	A repository-internal handle
Legal metadata	The applicable licence terms must be stated.

Objects with metadata from source systems or complex relations between data packages

Specification parameter	**Implementation**
Naming convention	The package is named at the top directory level with a unique identifier.

<table>
<tr><td><b>Package structure</b></td><td>
For each SIP there is a directory named with a unique identifier. This contains a dc.xml with Dublin Core metadata, optionally a harvest.xml with information about fetching from a data source and a collection.xml with information about assigning an object to a collection.

The representation directories contain the content files.

MD5 checksums can optionally be submitted as one checksum file for all files in all delivered identifier directories at the identifier directory level, or as one MD5 checksum per file in the representation directories.

Representation names apart from MASTER, PRE_INGEST_MODIFIED_MASTER and DERIVATIVE_COPY must be agreed with TIB.

```
root
|--[checksums].[md5] (optionally a checksum file for all
files in all delivered identifier directories)
|--IDENTIFIER
|        |--dc.xml (mandatory)
|        |--harvest.xml (optional)
|        |--collection.xml (optional)
|        |--MASTER (Pflicht)
|                |--1-n Files
|                |--0-1nFiles.[md5] (optionally one
checksum file per file in the representation)
|                |--0-n directories
|                        |-- ...
|        |--PRE_INGEST_MODIFIED_MASTER (optional)
|                |--1-n Files
|                |--0-n Files.[md5] (optionally one
checksum file per file in the representation)
|                |--0-n directories
|                |-- ...
|        |--DERIVATIVE_COPY (optional)
|                |--1-n Files
|                |--0-1nFiles.[md5] (optionally one
checksum file per file in the representation)
|                |--0-n directories
|                        |-- ...
|        |--SOURCE_MD (optional)
|                |--1-n .[xml]
|
|        |--more representations (optional)
|--IDENTIFIER
```

</td></tr>
<tr><td><b>Content data</b></td><td>at least one file per representation</td></tr>
<tr><td><b>Permissible file formats</b></td><td>No limitation</td></tr>
<tr><td><b>Representations</b></td><td>MASTER (mandatory), PRE_INGEST_MODIFIED_MASTER (optional), DERIVATIVE_COPY (optional), more representation by arrangement</td></tr>
<tr><td><b>Quality of data</b></td><td>Non-valid and non-well-formed files are also accepted.</td></tr>
<tr><td><b>Metadata</b></td><td>The object does not have to be indexed in the catalog. A dc.xml must be available.</td></tr>
<tr><td><b>Identifier</b></td><td>Unique identifier</td></tr>
</table>

Legal metadata	Legal metadata is captured in the CSV file as follows: (1) The access right to the document, as granted for use by the releasing entity, shall be documented via dcterms:accessRight (e.g., as private/public or with some other controlled vocabulary). (2) The archiving right of the releasing entity shall be documented as dc:rights . (3) The transfer agreement, as concluded between the TIB digital preservation team and the donating body. This is done via dcterms:license. 4) Access rights to the object in Rosetta are recorded as Access Right.

Pre-ingest SIPs

A submission application creates Rosetta-compliant pre-ingest SIPs from various transfer information packages, and transfers them to Rosetta during the second step.

Post-ingest SIPs

After deposit, the pre-ingest SIPs become post-ingest SIPs, which are enriched with additional metadata by the system. The transformation process is complete when a package has been transferred to permanent archival storage and successfully deposited there.

During further processing in Rosetta, the post-ingest SIP is transformed to an AIP, and is automatically enriched with additional metadata. A post-ingest SIP becomes an AIP once it has been transferred to permanent archival storage and saved there successfully.