

DocuteamDublinCore1.0 SIP format

Generalities

Definition

- A Docuteam DublinCore SIP is a .ZIP file containing a folder named `sip` which is a bagit container.
- The bagit must be created using at least sha256 checksums (other checksum algorithms supported by bagit are optional).
- Inside the bagit container, a hierarchical folder contains data objects described using XML DublinCore metadata.

References

- Bagit library:
 - <https://tools.ietf.org/id/draft-kunze-bagit-14.txt>
 - <https://github.com/LibraryOfCongress/bagit-spec>
 - <https://github.com/LibraryOfCongress/bagit-python>
 - <https://github.com/LibraryOfCongress/bagit-java>
- DublinCore
 - <http://dublincore.org/documents/dcmi-terms/>

bagit container structure specification

Within the zipped bagit, these SIP are organized as follows:

1. bagit contains **at least sha256 checksums**
2. the root folder, corresponding to the root object within the SIP, is named „data“ (this is handled automatically by bagit libraries)
3. subfolders may be named freely
4. subfolders may be organized recursively
5. in each folder (at all levels) there is a mandatory metadata file always named „dc.xml“
6. in addition, each folder (at all levels) may contain either (but not both!):
 - one or more subfolders
 - one datafile, which may be named freely (except „dc.xml“)

A more formal structure definition :

```
<rootfolder> ::= <metadata file> <children>*
<metadata file> ::= dc.xml
<children> ::= <folder>* | <file>
<folder> ::= <metadata file> <children>*
<file> ::= filename.ext
```

container structure examples

example 1 : container structure with only one file

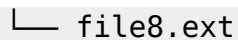
```
data/  
├─ dc.xml  
└─ filename1.ext
```

example 2 : container structure with several files

```
data/  
├─ dc.xml  
├─ folder1  
│   ├── dc.xml  
│   └─ fileA.ext  
├─ folder2  
│   ├── dc.xml  
│   └─ fileB.ext  
└─ folder3  
    ├── dc.xml  
    └─ fileC.ext
```

example 3 : complex structure with several files

```
data/  
├─ dc.xml  
├─ folder1  
│   ├── dc.xml  
│   ├── folder2  
│   │   ├── dc.xml  
│   │   └─ file3.ext  
│   └─ folder4  
│       ├── dc.xml  
│       └─ folder5  
│           ├── dc.xml  
│           └─ file5.ext  
├─ folder6  
│   ├── dc.xml  
│   └─ file6.ext  
└─ folder7  
    ├── dc.xml  
    └─ folder8  
        ├── dc.xml  
        └─ folder9  
            └─ dc.xml
```



metadata specification

Metadata is restricted to the Dublin Core Metadata Element Set, i.e. to 15 elements (dc 1.1 terms, see <http://dublincore.org/documents/dcmi-terms/#section-3>).

In addition, the following constraints apply:

1. The **„Identifier“ field is mandatory at each level in „dc.xml“, it must contain:**
 - **At each level: the the client application identifier** of the object with the prefix „clientid:“ e.g. „clientid:1234567“ or „clientid:d4FTw3v6T“
 - **At root level, a mandatory identifier with the customer namespace** in the repository (this is often the ISIL code) prefixed with „namespace:“, e.g. „namespace:CH-1234-1“
2. The **„Title“ field is mandatory at each level** in the „dc.xml“ file. It is not repeatable.
3. **All other 13 fields are optional and repeatable**, they are:
 - Creator (e.g. the authors, one per field repetition, that can be persons or institutions)
 - Subject (typically keywords, one per field repetition)
 - Description (a textual description of the object or folder)
 - Publisher
 - Contributor
 - Date (use ISO-8601, e.g. 2018-11-30)
 - Type
 - Format
 - Source
 - Language
 - Relation
 - Coverage
 - Rights

metadata examples

example 1: minimal metadata at root level

```
<?xml version="1.0" encoding="UTF-8"?>

<metadata
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <dc:title>Minimalist Example</dc:title>
  <dc:identifier>namespace:CH-123456-12</dc:identifier>
  <dc:identifier>clientid:12345</dc:identifier>

</metadata>
```

example 2: full metadata at root level

```
<?xml version="1.0" encoding="UTF-8"?>

<metadata
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
<dc:title>All fields are set</dc:title>
<dc:creator>Atreid, Leto</dc:creator>
<dc:creator>docuteam</dc:creator>
<dc:subject>dublincore</dc:subject>
<dc:subject>package</dc:subject>
<dc:subject>format</dc:subject>
<dc:description>Description of the docuteam dublin core package format,
version 1.0.</dc:description>
<dc:publisher>docuteam</dc:publisher>
<dc:contributor>Smith, John</dc:contributor>
<dc:contributor>Jaquard, Paul</dc:contributor>
<dc:date>2018-11-05</dc:date>
<dc:type>Text</dc:type>
<dc:format>application/pdf</dc:format>
<dc:identifier>namespace:CH-123456-12</dc:identifier>
<dc:identifier>clientid:999full</dc:identifier>
<dc:source>Dublin Core Package Structure
(https://docs.google.com/document/d/1lxqiqkmlNYVWlwJSsIe4b5DwJxN6DZqNvpo0MouAFIA/edit)</dc:source>
<dc:language>en</dc:language>
<dc:relation>docuteam bridge api for client applications
(https://docs.google.com/document/d/1GTHuk0lme\_fLlZZ-An8lEy2f2joMkjasHHAt0Asri\_0/edit)</dc:relation>
<dc:coverage>2018-2022</dc:coverage>
<dc:coverage>Baden</dc:coverage>
<dc:rights>CreativeCommons CC-By</dc:rights>

</metadata>
```

From:
<https://wiki.docuteam.ch/> - **docuteam wiki**

Permanent link:
https://wiki.docuteam.ch/doku.php?id=docuteam:sip_dc_public_documentation

Last update: **2019/11/05 08:26**

