

ORA objects in OCFL managed by Fedora6

Product description

An overview of how ORA4 objects will be stored in an OCFL-conformant file system using Fedora6. Alongside this document, the reader should have access to the ORA Data Model in JSON [ORA-JSON], and the sample ORA4 OCFL storage root [ORA-OCFL-ROOT]. In addition, this document should be read alongside the OCFL specification [OCFL-SPEC] and the Fedora6 documentation for OCFL Archival Groups in Fedora6 [FEDORA6-OCFL].

About this document

Version: 0.9

Author: Tom Wrobel <thomas.wrobel@bodleian.ox.ac.uk>

Created: 27 March 2019

Last modified: 24 May 2023

Background

As part of the ORA4 Digital Preservation Solution, ORA4 will seek to store a human-readable, stable, digital preservation copy of an object. The most appropriate format for this is in a file system that conforms to the Oxford Common File System specification (OCFL).

OCFL is “an application-independent approach to the storage of digital objects in a structured, transparent, and predictable manner. It is designed to promote long-term access and management of digital objects within digital repositories.” [OCFL-SPEC]. The OCFL specification describes the way in which multiple versions of an object are stored, and provides a set of rules which enable an object to be validated. The history of an object, including its current state, is stored in an `inventory.json` file in the object’s root.

ORA will use the Fedora6 repository application to manage the objects within its OCFL root. This means that objects will conform to the Fedora6 OCFL object structure for archival groups. The differences between an idealised ORA object in OCFL and changes necessary to use Fedora6 are discussed below (see ‘Fedora6 management files’)

OCFL concepts

OCFL has the concept of a ‘storage root’ directory. This storage root contains the OCFL objects, each in their own directory.

OCFL does not prescribe where object directories are stored, only that any directories between the storage root and an OCFL object are empty. However, it is advised that whatever the logical structure of an OCFL storage root, that the path from the storage root to an OCFL object is intelligible from its OCFL object identifier [OCFL-IMPLEMENTATION-NOTES].

In addition, OCFL makes a series of important requirements: OCFL mandates that once a file is written it is immutable, and that once an object version is created, that this version is immutable.

Fedora6 Management Files

As documented in [FEDORA6-OCFL], Fedora6 adds binary files with specific names and hierarchies to the OCFL object to manage the object within Fedora6.

Some of these files are used for Access Control Lists (ACLs): ORA will not be using ACLs, so these files will not be present.

The files and directories created by Fedora should be considered as ‘Fedora management files’ and can be ignored in any future use or versions of the object. Any non-Fedora software using an OCFL object created by Fedora6 should ignore these files completely: they are not relevant for the delivery of an ORA object, or any version of that object. It is possible that a future migration might remove these files from the active version of the object in Fedora. Alternatively, a future migration may choose to re-create the OCFL object as if these files were never present.

Allowed Fedora management files

- Fedora6 will add one binary file to the first version of an object. This binary file is called `fcr-container.nt` and is used by Fedora6 to store Fedora’s own object metadata triples for the object. It is stored in the content root for that version, i.e. `[object root]/v1/content` and will be an active file in the `inventory.json` of all subsequent versions.
- in each version’s content directory, Fedora6 will create a management directory at `‘./fcrepo’`. This directory its contents will be active files in the `inventory.json` of all subsequent versions. This directory and its contents are Fedora management files.

Any other Fedora6 management files present in the OCFL object system are not considered to be acceptable, and will mean that the OCFL object is not a valid ORA OCFL object.

Example structure (Fedora6 management files in red).

```
[object root]
├── 0=ocfl_object_1.1
├── inventory.json
└── inventory.json.sha512
  └── v1
    └── content
      ├── fcr-container.nt
      ├── .fcrepo
      │   ├── fcr-root.json
      │   ├── m1428cf48023972f0b24167801985d405~fcr-desc.json
      │   ├── m1428cf48023972f0b24167801985d405.json
      │   ├── mf03b51ef078f602f81577199944e4228~fcr-desc.json
      │   ├── mf03b51ef078f602f81577199944e4228.json
      │   └── uuid_e6443c52-39b1-4519-9e71-
      6e73df0ad020.metadata.ora.v2.json~fcr-desc.json
      |   |   └── uuid_e6443c52-39b1-4519-9e71-
      6e73df0ad020.metadata.ora.v2.json.json
      |   |   └── uuid_e6443c52-39b1-4519-9e71-
      6e73df0ad020.public_metadata.datacite.v4.xml~fcr-desc.json
      |   |   └── uuid_e6443c52-39b1-4519-9e71-
      6e73df0ad020.public_metadata.datacite.v4.xml.json
      |       ├── m1428cf48023972f0b24167801985d405
      |       ├── mf03b51ef078f602f81577199944e4228
      |       ├── uuid_e6443c52-39b1-4519-9e71-6e73df0ad020.metadata.ora.v2.json
      |       └── uuid_e6443c52-39b1-4519-9e71-
      6e73df0ad020.public_metadata.datacite.v4.xml
          ├── inventory.json
          └── inventory.json.sha512
```

ORA OCFL storage root and file system structure

ORA OCFL objects will use the Fedora6 OCFL storage root structure.

ORA OCFL objects will be stored in a directory corresponding to a SHA256 hash of their UUID prefixed with 'info:fedora/'. For an ORA object with the pid "uuid_abcdef01-abcd-abcd-abcd-abcdef013456", the ORA object directory will be a SHA256 hash of "info:fedora/uuid_abcdef01-abcd-abcd-abcd-abcd-abcdef013456", i.e.

"6fd2d0b8093abd453cde9f54a05bbfe84f861e7b5042c6e66e7ea0c087dfabee".

The path to this directory will be a pair-tree directory composed of the first three triplets of bytes in the directory. For example 6fd/2d0/b80/6fd2d0b8093abd453cde9f54a05bbfe84f861e7b5042c6e66e7ea0c087dfabee (see also [ORA-OCFL-ROOT]).

The storage root will contain an `ocfl_layout.json` file detailing the layout of the file system, as well as a copy of the final version of this document as a text file (`README.txt`) a copy of the ORA Data Model (`ora_data_model.xls`), and a copy of the ORA Data Model to JSON documentation (`ora-data-model-to-mets-2.3.js` [ORA-JSON]).

The storage root will contain an extension directory, adding conformance to OCFL extension 'Hashed N-Tuple storage layout' [STORAGE-LAYOUT] which defines the layout of the object pair-tree structure. The relevant `config.json` file will be the default file.

An ORA OCFL storage root containing one ORA4 object (uuid_abcdef01-abcd-abcd-abcd-abcdef013456) would therefore have the following structure:

```
└── extensions
    └── 0004-hashed-n-tuple-storage-layout
        └── config.json
└── ocfl_1.0.html
└── ocfl_layout.json
└── ora_datamodel_to_json-2.3.js
└── ora_data_model.xls
└── 6fd
    └── 2d0
        └── b80
            └── 6fd2d0b8093abd453c...
```

OCFL object structure

Each version of an OCFL object will contain a JSON file describing the object, its metadata, the binary files associated with the object, and their metadata. This JSON file will conform to the specification laid down in [ORA-JSON].

The JSON file will be named `{UUID}.metadata.ora.{datamodel-major-version}.json` e.g.
`uuid_abcdef01-abcd-abcd-abcd-abcdef013456.metadata.ora.v2.json`

Binary files will be saved with a filename corresponding to their fileset identifier, which is used in ORA as a binary file identifier as documented in [ORA-BINARY-FILE-IDENTIFIERS] e.g. 'r8s45q9350'

The JSON file will list file locations relative to the OCFL version's 'logical file path' (see [OCFL SPEC] Terminology): the JSON file will use this as the value of the `binary_files_file_path` value.

```
binary_files: [
    ...
    file_path: "file:///{{fileset_id}}"
]
```

In addition, a version of an OCFL object may contain a public metadata file in Datacite XML. This file will contain an openly disseminatable copy of the object's metadata. The file will be named {UUID}.public_metadata.datacite.{datacite-major-version}.xml e.g. `uuid_abcdef01-abcd-abcd-abcdef013456.public_metadata.datacite.v4.xml`

A simple ORA OCFL object (`uuid_e155a01a-5c6a-42cf-bb31-7fa2a6d553a7`) with a single binary file '`s7p88cg61f`' will therefore have the following structure (Fedora management files in red):

```
└── 0=ocfl_object_1.1
    ├── inventory.json
    ├── inventory.json.sha512
    └── v1
        ├── content
        │   ├── fcr-container.nt
        │   └── .fcrepo
        │       ├── fcr-root.json
        │       ├── s7p88cg61f~fcr-desc.json
        │       ├── s7p88cg61f.json
        │       ├── uuid_e155a01a-5c6a-42cf-bb31-
        │       │   7fa2a6d553a7.metadata.ora.v2.json~fcr-desc.json
        │       └── uuid_e155a01a-5c6a-42cf-bb31-
        │           7fa2a6d553a7.metadata.ora.v2.json
        │               ├── s7p88cg61f
        │               │   └── uuid_e155a01a-5c6a-42cf-bb31-7fa2a6d553a7.metadata.ora.v2.json
        │               └── inventory.json
        └── inventory.json.sha512
```

A multiple versioned ORA OCFL object will a version directory for each version. Each version MUST contain an ORA JSON metadata file describing that version. In the following example:

1. An initial object (`uuid_3ebe76fd-ee81-4f59-9fcf-4c395d975b3f`) was created with one file, `mc4702b8d33c212636def1e26af9a1dd1`
2. An additional file was subsequently deposited, `r8s45q9350` and the object was published. This added a public metadata file to the record.
3. For brevity, the `.fcrepo` directories have been omitted from complex examples

```
└── 0=ocfl_object_1.1
    ├── inventory.json
    ├── inventory.json.sha512
    └── v1
        ├── content
        │   ├── fcr-container.nt
        │   └── mc4702b8d33c212636def1e26af9a1dd1
        │       └── uuid_3ebe76fd-ee81-4f59-9fcf-4c395d975b3f.metadata.ora.v2.json
        ├── inventory.json
        └── inventory.json.sha512
    └── v2
        ├── content
        │   ├── r8s45q9350
        │   └── uuid_3ebe76fd-ee81-4f59-9fcf-4c395d975b3f.metadata.ora.v2.json
        └── inventory.json
            └── inventory.json.sha512
4c395d975b3f.public_metadata.datacite.v4.xml
```

Sample objects

A suite of sample objects will be maintained in the ORA Data Model github project [ORA-OCFL ROOT].

References

[FEDORA6-OCFL]

URL: <https://wiki.lyrasis.org/display/FEDORA6x/Fedora+OCFL+Object+Structure>

[OCFL-IMPLEMENTATION-NOTES]

Hankinson, A. et al. (2020), *Implementation notes*. URL: <https://ocfl.io/1.0/implementation-notes/>

[OCFL-SPEC]

Hankinson, A. et al. (2020), *Oxford Common File Layout Specification 1.0*. URL: <https://ocfl.io/1.0/spec/>

[ORA-BINARY-FILE-IDENTIFIERS]

URL: <https://gitlab.bodleian.ox.ac.uk/ORA4/ora4-developers/-/wikis/development/plans/Unique-binary-file-ids>

[ORA-JSON]

Wrobel, T. (2022), *The ORA Data Model in JSON*, URL: https://github.com/bodleian/ora_data_model/blob/master/ora_datamodel_to_json-2.3.js

[ORA-METS]

Wrobel, T. (2022), *The ORA Data Model in METS*, URL: https://github.com/bodleian/ora_data_model/blob/master/ora_datamodel_to_mets-2.3.xml

[ORA-OCFL-ROOT]

URL: https://gitlab.bodleian.ox.ac.uk/ORA4/ora_ocfl/-/tree/master/storage_root

[STORAGE-LAYOUT]

URL: <https://ocfl.github.io/extensions/0004-hashed-n-tuple-storage-layout.html>