Fedora OCFL Object Structure

Layout of a Fedora OCFL Object

This document shows a series of diagrams depicting the structure of the three varieties of Fedora resources compared with their non-Fedora OCFL representations. Following the diagrams are detailed descriptions of the Fedora naming conventions.

The three varieties of Fedora resources are:

1. **Atomic container**: A Fedora container resource that maps to a single OCFL object
2. **Atomic binary**: A Fedora binary resource that maps to a single OCFL object
3. **Archival group**: A potentially hierarchical set of Fedora resources (binaries and/or containers) that maps to a single OCFL object
    a. Note, an archival group is comprised of **Archival parts**

Finally, a detailed description is provided of the internal contents of the Fedora-specific files.

For more information on OCFL, see the specification.

This document is intended to provide a "logical" description of what may be found when exploring Fedora's OCFL storage root. What is actually on-disk may be different based on details of the OCFL specification. For example:

- Identical files in the same object will only be persisted once, and referenced in the OCFL inventory.json as multiple files.
- Each subsequent OCFL version directory will only contain the changed/added files, as opposed to the complete set of files for the OCFL object. This is the concept of "forward versioning".
- Fedora installations that choose to disable auto-versioning will see "mutable head" directories in their OCFL objects.

To provide the most intuitive view of your OCFL storage root, it is recommended that an OCFL client be used to explore the files.

General

In the examples below, it can be seen that OCFL objects created by Fedora consist of three types of files:

1. The user-content files
2. RDF files containing user-properties related to the OCFL object and/or user's content files
3. Fedora-specific JSON "header" files containing system-properties related to the OCFL object and/or user's content files

The first two categories of files (user's content files and Fedora-specific RDF files) are located in the "content/" section of the OCFL object. The last category of files (Fedora-specific JSON "header" files) are located in a sub-directory of the "content/" section of the OCFL object, named ".fcrepo/".

The full contents and format of the JSON "header" files, and Fedora-specific RDF files are detailed in the Specification of Fedora-specific OCFL Files section of this document.

Atomic Resources

An atomic resource is either a single binary or a single container that maps to its own OCFL object. Below is a comparison between a plain OCFL representation of a single item object and Fedora's atomic resource representation.

Single-binary OCFL Object

The following OCFL Object has a single version, which holds a single binary: "image.tiff":

```
[object root]
     0=ocfl_object_1.0
     inventory.json
     inventory.json.sha512
     v1/
         inventory.json
         inventory.json.sha512
         content/
             image.tiff
```

Fedora Atomic Resource - Binary

The following is the same single-binary OCFL object with both optional and required files used by Fedora.

Note: the ACL resources depicted in this and the following diagrams may be omitted if the given resource chooses to inherit its ACL from the repository's default ACL.

```
[object root]
     0=ocfl_object_1.0
     inventory.json
     inventory.json.sha512
     v1/
         inventory.json
         inventory.json.sha512
         content/
             .fcrepo/
                fcr-root.json           <-- Required "header" file holding system metadata about the binary.
See description below.
                fcr-root~fcr-desc.json  <-- Required "header" file holding system metadata about the binary's
description. See description below.
                fcr-root~fcr-acl.json   <-- Optional, only present if this Fedora resource has its own ACL.
             image.tiff
             image.tiff~fcr-desc.nt      <-- Required "binary description". See description below.
             image.tiff~fcr-acl.nt       <-- Optional, only present if this Fedora resource has its own ACL.
```

Fedora stores system metadata about the binary (image.tiff) and the binary's description (currently, https://host/rest/image.tiff/fcr:metadata) in respective JSON files that contain elements such as the creation date and creator, as well as its interaction model (i.e. type), whether the object is an archival group, etc. Additionally, Fedora resources can optionally have their own ACLs, which if present also have their own JSON header file.

Naming conventions

System content (located in the ".fcrepo/" directory)

- The *binary's header file*: the filename is always "fcr-root.json". This allows Fedora to find the header file without knowing if the resource is a binary, container or archival group.
- The *binary description's header file*: the filename is always "fcr-root~fcr-desc.json".
- The *binary ACL's header file*: the filename is always "fcr-root~fcr-acl.json".

User content

- The *binary*: the filename provided during ingest of the binary is retained in the OCFL persistence. ("image.tiff", in this example)
- The *binary's description*: the filename is the name of the binary with the addition of the suffix, "~fcr-desc.nt". ("image.tiff~fcr-desc.nt", in this example)
- The *binary's ACL*: the filename is the name of the binary with the addition of the suffix, "~fcr-acl.nt". ("image.tiff~fcr-acl.nt", in this example)

Fedora Atomic Resource - Container

A Fedora container resource is effectively an empty directory with LDP-defined behaviors depending on its "interaction model" (i.e. type: basic, direct, indirect), and associated metadata, along with an optional ACL. Since OCFL does not support empty directories as valid objects, there is no direct OCFL analogous object for a Fedora container atomic resource.

Note: the ACL resources depicted in this and the following diagrams may be omitted if the given resource chooses to inherit its ACL from the repository's default ACL.

```
[object root]
     0=ocfl_object_1.0
     inventory.json
     inventory.json.sha512
     v1/
         inventory.json
         inventory.json.sha512
         content/
             .fcrepo/
                fcr-root.json          <-- Required "header" file holding system metadata about the container.
See description below.
                fcr-root~fcr-acl.json  <-- Optional, only present if this Fedora resource has its own ACL.
             fcr-container.nt           <-- Required file for holding user-properties describing the container.
See description below.
             fcr-container~fcr-acl.nt   <-- Optional, only present if this Fedora resource has its own ACL.
```

Naming conventions

System content (located in the ".fcrepo/" directory)

- The *container's header file*: the filename is always "fcr-root.json". This allows Fedora to find the header file without knowing if the resource is a binary, container or archival group.
- The *container ACL's header file*: the filename is always "fcr-root~fcr-acl.json".

User content

- The *container user-properties*: the filename is always "fcr-container.nt".
- The *container's ACL*: the filename is always "fcr-container~fcr-acl.nt".

Archival Group Resources

A Fedora archival group resource is container that contains a potentially nested hierarchy of zero or more children containers and/or binaries. Below is a comparison between a plain OCFL representation of a compound object and Fedora's archival group resource representation.

Compound OCFL Object

The following is an example of an OCFL object that contains one version, consisting of two binaries, one of which is nested within a child container.

```
[object root]
     0=ocfl_object_1.0
     inventory.json
     inventory.json.sha512
     v1/
         inventory.json
         inventory.json.sha512
         content/
             image.tiff
             foo/
                 bar.xml
```

Fedora Archival Group

Archival groups are a way to collect several Fedora resources into a single OCFL object. The constituent Fedora resources (archival parts) can be any combination of containers and binaries.

Note: the ACL resources depicted in this diagram may be omitted if the given resource chooses to inherit its ACL from an ancestor resource in the nested hierarchy or even from the repository's default ACL.

```
[object root]
     0=ocfl_object_1.0
     inventory.json
     inventory.json.sha512
     v1/
         inventory.json
         inventory.json.sha512
         content/
             .fcrepo/
                fcr-root.json              <-- Required "header" file holding system metadata about the
archival group.
                fcr-root~fcr-acl.json      <-- Optional, only present if this Fedora resource has its own ACL.
                image.tiff.json            <-- Required "header" file holding system metadata about the binary.
                image.tiff~fcr-desc.json   <-- Required "header" file holding system metadata about the
binary's description.
                image.tiff~fcr-acl.json    <-- Optional, only present if this Fedora resource has its own ACL.
                foo.json                   <-- Required "header" file holding system metadata about the nested
container.
                foo~fcr-acl.json           <-- Optional, only present if this Fedora resource has its own ACL.
                foo/                       <-- Required nested structure within .fcrepo/ mirrors content
structure
                    bar.xml.json           <-- Required "header" file holding system metadata about the binary.
                    bar.xml~fcr-desc.json  <-- Required "header" file holding system metadata about the
binary's description.
                    bar.xml~fcr-acl.json   <-- Optional, only present if this Fedora resource has its own ACL.
             fcr-container.nt               <-- Required file for holding user-properties describing the
archival group container.
             fcr-container~fcr-acl.nt       <-- Optional, only present if this Fedora resource has its own ACL.
             image.tiff
             image.tiff~fcr-desc.nt         <-- Required "binary description".
             image.tiff~fcr-acl.nt          <-- Optional, only present if this Fedora resource has its own ACL.
             foo/
                 fcr-container.nt           <-- Required file for holding user-properties describing the
archival part container.
                 fcr-container~fcr-acl.nt   <-- Optional, only present if this Fedora resource has its own ACL.
                 bar.xml
                 bar.xml~fcr-desc.nt        <-- Required "binary description".
                 bar.xml~fcr-acl.nt         <-- Optional, only present if this Fedora resource has its own ACL.
```

Naming conventions

System content (located in the ".fcrepo/" directory)

- The *archival group container's header file*: the filename is always "fcr-root.json". This allows Fedora to find the header file without knowing if the resource is a binary, container or archival group.
- The *archival group container ACL's header file*: the filename is always "fcr-root~fcr-acl.json".
- The *archival part container's header file*: the filename is the name of the archival part container with the addition of the ".json". ("foo.json" in this example)
- The *archival part container ACL's header file*: the filename is the name of the archival part container with the addition of the suffix, "~fcr-acl.json". ("foo~fcr-acl.json" in this example)
- The *binary's header file*: the filename is the name of the binary with the addition of the ".json" extension. ("image.tiff.json" and "foo/bar.xml.json" in this example)
- The *binary description's header file*: the filename is the name of the binary with the addition of the suffix, "~fcr-desc.json" extension. ("image.tiff~fcr-desc.json" and "foo/bar.xml~fcr-desc.json" in this example)
- The *binary ACL's header file*: the filename is the name of the binary with the addition of the suffix, "~fcr-acl.json" extension. ("image.tiff~fcr-acl.json" and "foo/bar.xml~fcr-acl.json" in this example)

User content

- The *archival group container user-properties* and *sub-container user-properties*: the filename is always "fcr-container.nt".
- The *archival group container's ACL* and *sub-container's ACL*: the filename is always "fcr-container~fcr-acl.nt".
- The *binary*: the filename provided during ingest of the archival part binary is retained in the OCFL persistence. ("image.tiff" and "foo/bar.xml" in this example)
- The *binary's description*: the filename is the name of the binary with the addition of the suffix, "~fcr-desc.nt". ("image.tiff~fcr-desc.nt" and "foo/bar.xml~fcr-desc.nt" in this example)
- The *binary's ACL*: the filename is the name of the binary with the addition of the suffix, "~fcr-acl.nt". ("image.tiff~fcr-acl.nt" and "foo/bar.xml~fcr-acl.nt" in this example)
- The *sub-container(s)*: the filename provided during ingest of the archival part container is retained in the OCFL persistence. ("foo/" in this example)

Storage Hierarchy Layout

The content that Fedora persists (user and system files) is a complete representation of the repository. These files are written to OCFL objects within a top-level storage hierarchy. This section describes the storage hierarchy layout and the algorithm for mapping an OCFL object identifier to its storage hierarchy path.

Unless configured otherwise, Fedora uses the default configuration of the "OCFL Community Extension 0003: Hashed Truncated N-tuple Trees for OCFL Storage Hierarchies" (0003-hashed-n-tuple-trees). That configuration is:

```
{
    "digestAlgorithm": "sha256",
    "caseMapping": "toLower",
    "tupleSize": 3,
    "numberOfTuples": 3,
    "shortObjectRoot": false
}
```

The SHA-256 hash of a Fedora resource's ID is calculated, converted to lowercase, then split into three-character segments. The first three segments are used in creating the first three levels of directories below the OCFL storage root. The OCFL object is persisted within a fourth-level directory, which is the entire, lowercase, SHA-256 value previously calculated. The extension definition provides an example.

Specification of Fedora-specific OCFL Files

Every Fedora OCFL object contains a ".fcrepo/" directory in the root of its "content/" directory. This directory contains JSON "header" files that describe the resources contained in the OCFL object. They contain information that enables Fedora to know how it is supposed to interact with a resource. Header files must be identifiable by Fedora without knowing anything about a resource other than its ID.

Outside of the ".fcrepo/" directory, all other OCFL object files store the content of Fedora resources. Every Fedora resource is persisted to OCFL using exactly two files. The first is a header file located in ".fcrepo/" and the second is a content file located in the root of the OCFL object's "content/" directory. The naming and location of content files can vary depending on the type of resource they belong to. It is not necessarily possible to identify a resource's content file without first determining what type of resource it is by reading its header file.

RDF Files

Fedora creates RDF files for three purposes:

1. For persisting WebACLs associated with a given Fedora resource
2. For persisting user-provided descriptions of binary resources
3. For persisting user-provided properties associated with container resources

ACLs

The RDF content of WebACLs is user-provided, and must conform to the Web Access Control specifications. WebACL files are optional, and will only exist if the user creates an ACL on a Fedora resource.

Binary Descriptions

The RDF content of binary descriptions is user-provided, and can be any valid RDF. By convention, the content of the RDF should be used to describe the associated binary resource. Binary description files must exist for each binary resource and are auto-created by Fedora as empty files when a binary resource is created.

Container Properties

The RDF content of container resources is user-provided, and can be any valid RDF. Container property files must exist for each container resource and are auto-created by Fedora as empty files when a container resource is created.

Header Files

For a full description of the contents of Fedora header files, see Fedora Header Files.