

2020-04-29 - Fedora Leaders - Committers Meeting

Time/Place

Time: 11:00am Eastern Time (US)

Please see calendar invite for Zoom link.

Attendees

1. [Andrew Woods](#)
2. [David Wilcox](#)
3. [Danny Bernstein](#) ★
4. [Thomas Bernhart](#)
5. [Jon Dunn](#)
6. [Raman Ganguly](#)
7. [Scott Prater](#)
8. [Ben Pennell](#)
9. [Peter Winckles](#)
10. [Jennifer Gilbert](#)
11. [Doron Shalvi](#)
12. [Jared Whiklo](#)
13. [Robin Lindley Ruggaber](#)
14. [Este Pope](#)
15. [Mark Jordan](#)
16. [Dan Field](#)

Agenda

Open development questions

1. To what extent should Fedora 6 support vanilla OCFL? i.e:
 - a. Do we value Fedora being able to run on top of vanilla OCFL? - in a read-only fashion? in a read/write fashion?
 - b. Do we value a post-Fedora OCFL that is not peppered with Fedora-specific system metadata?

Use cases

1. As a systems integrator, I would like a simple GET/PUT HTTP server over OCFL
2. As a Fedora 3 repository manager, I would like to export my content as vanilla OCFL
3. As a repository manager, I would like to bring my vanilla OCFL into Fedora
4. As a digital preservationist, I would like to keep the OCFL created by Fedora but stop using Fedora

Context

1. In order for Fedora to manage repository content, Fedora-specific system metadata is required (caveat for vanilla OCFL)
2. Although it is meaningful and self-describing, there is some stakeholder concern around including content in an OCFL object that is in addition to what the user provided
3. Currently, Fedora segregates this system metadata into its own directory (".fcrepo/") within the OCFL object
 - a. What system metadata are we exactly talking about?

```

# all Fedora resources have:
"parent":"info:fedora/new",
"id":"info:fedora/new/child",
"interactionModel":"http://www.w3.org/ns/ldp#BasicContainer",
"createdDate":"2020-04-21T16:52:42.757566Z",
"lastModifiedDate":"2020-04-21T16:52:42.757566Z",
"createdBy":"","
"createdDate":"","
"stateToken":"A1FAFA3B80B37AD8C1B0CC519ABD30A2",
"archivalGroup":false,
"objectRoot":true

# binaries also have:
...
"filename":"notes.txt",
"mimeType":"text/plain",
"contentSize":282,
"digests":[],
...

# external content also has:
...
"externalUrl":,
"externalHandling":
...

```

b. Structure and location of Fedora system metadata in an OCFL object

```

test-object
  0=ocfl_object_1.0
  inventory.json
  inventory.json.sha512
  v1/
    inventory.json
    inventory.json.sha512
    content/
      .fcrepo/
        test-object.json
        test-object.nt
  v2/
    inventory.json
    inventory.json.sha512
    content/
      .fcrepo/
        empty-txt-description.json
        empty-txt.json
  v3/
    inventory.json
    inventory.json.sha512
    content/
      .fcrepo/
        bar.xml-description.json
        bar.xml.json
      foo/
        bar.xml
  v4/
    inventory.json
    inventory.json.sha512
    content/
      .fcrepo/
        image.tiff-description.json
        image.tiff.json
      image.tiff

```

Notes

The goals for this meeting include:

1. Establish shared understanding of Fedora 6's relationship with "vanilla" OCFL
2. Gain consensus on expectations/requirements going forward, with respect to Fedora 6 and OCFL

Discussion

1. The OCFL Fedora generates is OCFL compliant.
2. There have been stakeholders that have defined "Plain Vanilla" as including only user content - no application specific metadata.
3. We can talk about stripping out .fcrepo directories - but is this a real use-case? Does anyone really need that capability?
 - a. We should likely specify the meaning of any files in .fcrepo (using JSON Schema or the like) so that future applications (post-Fedora) can make use of the .fcrepo.
 - b. In a post-Fedora world, .fcrepo stuff is converted to new application stuff to what extent can it deal with .fcrepo stuff in previous versions.
 - c. We could support exporting to a new OCFL without the .fcrepo sprinkles.
 - d. Is JSON the right format for the metadata? Should we consider PREMIS xml? Or is a clear specification with a mapping to other formats sufficient? - likely a schema and cross-walk to other standards is sufficient
4. Additional use cases
 - a. Migration from Fedora 6 to a future Fedora (e.g. Fedora 10): will we support this? Will we ensure that there is compatibility going forward between Fedora versions ? YES!
 - b. As a repository manager, I want to add content directly to the OCFL managed by Fedora ("side-load"). Although not discussed, this has previously been noted as a "bulk ingest" use case
5. Caveats around Plain Vanilla (PV) OCFL:
 - a. Fedora could manage PV with certain assumptions
 - b. Discoveries, Observations, Opportunities if we want to support read-write with PV OCFL:
 - i. Assumptions
 1. OCFL objects not authored by Fedora 6 must be treated as a collection of binaries
 2. You can add new binaries
 3. Relationships would not be supported between objects beyond inherent "ldp:contains" relationships
 4. Binary descriptions would not necessarily exist
 - ii. Opportunity
 1. Supporting this more general use-case would expand the relevance of Fedora
 - iii. What do we want to support?
 1. Supporting Plain Vanilla and Fedora with Sprinkles simultaneously in a single repo may add significant complexity
 2. However because OCFL is immutable don't we need to support the mixed case (The Swirl), no? - unless PV objects are converted into F6 objects. The Swirl would be needed to properly read and serve previous versions of the object
6. The major advantage of OCFL is transparency, not "purity" (ie freedom from any application specific information).

Outstanding questions

1. Do we have general agreement on the following points?
 - a. Fedora should support being layered over vanilla OCFL objects
 - b. In order to provide the value-added features of Fedora 6, Fedora-specific metadata will need to be added to OCFL objects
2. Can we support Fedora/OCFL objects that come into Fedora as vanilla, but subsequently have Fedora 6 versions?

Action Items

- Type your task here, using "@" to assign to a user and "/" to select a due date