Fedora and the Oxford Common File Layout

Enhancing Support for Digital Preservation

David Wilcox
DuraSpace,Canada
dwilcox@duraspace.org
0000-0001-5411-9208

Abstract – Fedora is an open source repository platform for managing and preserving digital objects. While Fedora has always been associated with digital preservation, recent releases have focused on exposing linked data and aligning with modern web standards. The Oxford Common File Layout (OCFL), which defines a shared approach to file hierarchy for long-term preservation, provides an opportunity to bring the focus back to digital preservation in Fedora. The OCFL supports application-independent, transparent file persistence that can be used to rebuild a repository in case of disaster. These features address the current needs of the Fedora community, so a group of Fedora committers met in person to design a version of Fedora that implements the OCFL. This will be the focus of the next major release, Fedora 6. 0. This paper introduces the OCFL and describes the proposed design for Fedora 6. 0, including the next steps for development and implementation.

Keywords – Fedora, repository, OCFL, preservation, standards

Conference Topics – Designing and Delivering Sustainable Digital Preservation; The Cutting Edge: Technical Infrastructure and Implementation

I. Introduction

Fedora [1] is a flexible, extensible, open source repository platform for managing, preserving, and providing access to digital content. For the past several years the Fedora community has prioritized alignment with linked data best practices and modern web standards. However, the focus is now shifting back to Fedora's digital preservation roots with the help of the Oxford Common File Layout (OCFL) [2]. The OCFL, which began as a discussion at the Fedora and Samvera Camp at Oxford, UK in 2017, is an effort to define a shared approach to file hierarchy for long-term preservation. This approach includes both transparency and the ability to rebuild a repository from the contents on disk, both of which address key needs in the Fedora community. With the support of the Fedora governance group, a team of Fedora committers met in person in February of 2019 to design the next major release of Fedora, which will implement the OCFL at the persistence layer. This paper introduces the OCFL and describes the proposed design for Fedora 6. 0, including the next steps for development and implementation.

II. The Oxford Common File Layout

The Oxford Common File Layout (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. ” [3]. The OCFL arose from the need to insulate digital objects, which tend not to change much after being accessioned, from the churn of software applications, which tend to change much more often. These application updates often involve data migrations, which put the data at risk. They also store data in application-dependent ways, making it difficult or impossible to understand the data without the software. The OCFL mitigates these issues by specifying a file and directory layout that applications must conform to.

The most basic element of the specification is the OCFL Object, which is “a group of one or more content files and administrative information, that are together identified by a URI. The object may contain a sequence of versions of the files that represent the evolution of the object's contents. ” [3]. OCFL Objects contain administrative information that identifies them as OCFL Objects and allows changes to be tracked over time. The content files of an OCFL

16th International Conference on Digital Preservation
iPRES 2019, Amsterdam, The Netherlands.

Object can be anything at all; however, in order to support rebuilding the repository from the files on disk, OCFL Objects should contain "all the descriptive, administrative, structural, representation and preservation metadata relevant to the object. " [3].

An OCFL Object contains a file declaring its conformance with a particular version of the specification, along with a sequentially numbered folder for each version of the content files. A version folder (e. g. v1) contains a content folder (if it has contents), an inventory file, and an inventory digest file. The inventory file is a JSON document with a manifest of all the files in the version's content folder. Each file in an OCFL Object has an associated digest, which both provides a fixity value that guards against degradation over time and allows for a content-addressable reference to the file.

III. Motivations for Implementing OCFL In Fedora

Fedora is a digital repository for the long-term storage and management of digital objects. Fedora has gone through several upgrades over the years as the needs of its user community changed and technologies improved. Some of these upgrades have required data migrations, which makes them very challenging for institutions to absorb, especially if they have large amounts of data. It also puts the data at risk of corruption during the move. By making Fedora OCFL-compliant, future upgrades should not require data migrations. This is because the application will be made to conform with the files on disk, rather than the other way around.

Another motivation is transparency. Currently, Fedora objects are stored in a database and file structure that is application-dependent, meaning the contents of the repository cannot be inspected and understood without going through the Fedora application. This presents a risk to future access; if a hard drive with the contents of the repository were recovered without the Fedora application layer, the contents would be difficult to interpret.

Finally, the ability to rebuild the repository from the files on disk is an important motivator. Currently, backup and restore tools must be used to recover the repository in the case of a problem. This new functionality would allow the repository to be rebuilt by reading the contents of the files on disk. For all of these reasons,, the Fedora community has decided to implement the OCFL in the next major version of Fedora.

IV. Designing Fedora 6. 0

A. *Implementing OCFL*

A group of Fedora committers met in-person in February, 2019 to design Fedora 6. 0, the next major version of the software. The team went into the meeting with several design goals:

1. Implement the OCFL in Fedora
2. Improve performance and scale
3. Support complete repository rebuilds from the contents on disk
4. Don't make major changes to the API

With these goals in mind, the team discussed how best to implement the OCFL in Fedora in a way that would be scalable and performant without causing undue problems for users who might have written applications against the API. The first problem to address was the mapping between Fedora objects and OCFL objects, which are not exactly the same. Specifically, Fedora objects are based on the Linked Data Platform (LDP) specification [4], and contain Fedora-specific information. In the interests of backward compatibility, scale, and the ability to rebuild the repository from the file system, the resulting objects will contain fedora-specific metadata and may be required to follow specific naming conventions, and have other structural requirements placed on them.

In the interests of archival transparency, an opt-in extension to the Fedora API will allow grouping of resources that are persisted and versioned together as a single archival unit. All resources underneath an "Archive group" container will be persisted within a single archive (i. e. within an OCFL object). This will allow for the creation and maintenance of compound OCFL objects containing several files within them.

It will also be possible to drop an instance of Fedora on top of an existing OCFL storage root; in this case, Fedora will be able to read and make sense of the contents of the file system. The existing OCFL data are not required to contain any fedora-specific metadata, or follow any specification, convention, or be otherwise related to Fedora in any way. This approach has many advantages, one of which is to create a plausible migration path from Fedora 3. x by converting the contents on disk to be OCFL-compliant before dropping Fedora 6. x on top.

B. Other Features

In addition to implementing the OCFL, Fedora 6. 0 will also include a number of other features and improvements. One of these will be a built-in query endpoint for simple, common repository queries. Since version 4. 0, Fedora has not supported an internal query service, instead delegating such functionality to external tools like Apache Solr. However, the community has expressed a need for a synchronous, internal query service, so this will be added in Fedora 6. 0. The supported queries will include:

1. List all resources
2. List resources by mimetype
3. List resources by parent
4. List resources by mimetype, parent, and modified date (<>=)
5. List resources where modified <> x date.

Users will still need to use an application like Solr for more complex queries, but there is already an out-of-the-box integration based on Apache Camel that can be set up and used with a standard Fedora installation.

While Fedora currently supports fixity checking, the community has expressed a need for a more robust, proactive fixity service. This new service will automatically check the fixity of all items in the repository at a frequency and schedule specified by the administrator. It will log the results and report errors, and it will also maintain a full report of the health of the repository that can be requested on-demand.

C. Architecture

Fedora 6. 0 will be architected to support greater performance and scale while complying with the OCFL. This will be achieved by replacing the current ModeShape backend with an OCFL-compliant file system while optimizing reads and lookups with an internal database. This database will act as a kind of fast cache on top of the relatively slow filesystem.

V. Next Steps

Following the design meeting in February, the Fedora committers put together a high level summary of the design to share, first with the Fedora Leadership Group, and then with the broader Fedora community. The goal of this effort is to get buy-in, both from the Fedora governance group and the community as a whole.

Once we have buy-in from the Fedora governance group and the community, we will proceed to schedule code sprints to complete the work. While the Fedora project has full-time staff through its relationship with DuraSpace, these staff members do not write the majority of the code for the software, instead playing roles as community and technical coordinators. The bulk of the code and documentation will be written by members of the community, which is why achieving buy-in is important. We will also get commitments from institutions to adopt Fedora 6. 0 when it is ready, and we will work with these institutions as we develop the software to ensure we are building the application that the community wants.

VI. Conclusion

Over the years, the Fedora community has prioritized and focused on different aspects of the software. The 4. x line of Fedora releases put the emphasis on support for linked data and alignment with modern web standards. This culminated with the release of Fedora 5. 0, which implements the recently completed Fedora API specification [5]. Having reached this milestone, the community has returned to a focus on digital preservation, which coincides with the development of the OCFL specification. The OCFL represents a return to the digital preservations sensibilities of Fedora 3. x, but as a more standardized, community-focused effort. With the completion of the initial design of Fedora 6. 0, the community will proceed to put together a development plan, including a combination of code sprints and funded development effort. We are targeting late 2019 for the 6. 0 release, which will bring together the linked data and web standard features with the strong digital preservation sensibilities of the OCFL.

Acknowledgement

This paper would not have been possible without the support of the Fedora community, particularly those institutions that are members of DuraSpace in support of Fedora.

References

[1] Fedora - The Flexible, Modular, Open Source Repository Platform. [Online]. Available: https://duraspace. org/ fedora/. [Accessed March 15, 2019].

[2] Oxford Common File Layout. [Online]. Available: https:// ocfl. io. [Accessed March 15, 2019].

[3] A. Hankinson, et al, "Oxford Common File Layout Specification 0. 2," March 12, 2019. [Online]. Available: https://ocfl. io/0. 2/spec/. [Accessed March 15, 2019].

[4] S. Speicher, J. Arwe, A. Malhotra, "Linked Data Platform 1.0," February 26, 2015. [Online]. Available: https://www.w3.org/TR/ldp/. [Accessed March 19, 2019].

[5] B. Armintor, E. Cowles, D. Lamb, S. Warner, A. Woods, "Fedora API Specification 1.0," November 22, 2018. [Online]. Available: https://fedora.info/2018/11/22/spec/. [Accessed March 19, 2019].