Roadmap

2019-2020 Technical Roadmap

Fedora 6.0.0

The next major version of Fedora will focus on the following requirements:

1. Replace the ModeShape persistence layer with a different technology that implements the Oxford Common File Layout
2. Add a synchronous query service
3. Improve the fixity service
4. Address known performance and scale issues
5. Support migrations from earlier versions of Fedora (3.x, 4.x, and 5.x)

Further details can be found on the design page.

Fedora 6 development is expected to take place over the course of monthly code sprints throughout 2020.

Why the Oxford Common File Layout?

The OCFL provides the following benefits:

1. **Parsability**, both by humans and machines, to ensure content can be understood in the absence of original software
2. **Robustness** against errors, corruption, and migration between storage technologies
3. **Versioning**, so repositories can make changes to objects allowing its history to persist
4. **Storage diversity**, to ensure content can be stored on diverse storage infrastructures including cloud object stores
5. **Completeness**, so that a repository can be rebuilt from the files it stores

These benefits supplement the digital preservation features already provided by Fedora, including:

1. **Fixity**: Checksums can be calculated, stored and compared on demand
2. **Versioning**: Objects and files can be versioned and restored on demand
3. **Import/Export**: Objects and files can be exported on demand to facilitate their use in other elements of a digital preservation workflow
4. **Audit**: Preservation metadata can be generated by repository events and indexed in a triplestore for querying

The combined functionality of Fedora with OCFL persistence will better support an overall digital preservation strategy.

2017-2018 Technical Roadmap

Formalize the core Fedora services Application Programming Interface (API)

This priority is to clearly define the core services that Fedora promises as a standards-based RESTful API, accompany this API with any necessary domain-specific ontologies, and provide a compatibility test suite. Outstanding issues can be found on GitHub.

Align the current Fedora implementation with the API specification

Once the API specification is complete, the current Fedora implementation will need to be updated to fully align with the specification. This work will result in a 5.x Fedora release based on our move to semantic versioning.

Support alternate Fedora implementations

One of the goals of the API specification is to allow the community to experiment with different back-end Fedora implementations to address different use cases. We will support and encourage community members as they experiment along these lines.

2016-2017 Technical Roadmap

1. Formalize the core Fedora services Application Programming Interface (API)
   This priority is to clearly define the core services that Fedora promises as a standards-based RESTful API, accompany this API with any necessary domain-specific ontologies, and provide a Technology Compatibility Kit (TCK) for each service.

type	key	summary	assignee	reporter	priority	status	resolution	created	updated	due

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

The Fedora services are:

a. Create/Read/Update/Delete on repository resources
    i. Standard: Linked Data Platform
    ii. Include Import and Export of RDF, and option for RDF serialization to disk
    iii. type key summary assignee reporter priority status resolution created updated du

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

b. Versioning
    i. Standard (partial, only retrieval): Memento
    ii. type key summary assignee reporter priority status resolution created updated du

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

c. Atomic Batch Operations
    i. Standard: TBD
    ii. key summary type created updated due assignee reporter priority status resolutic

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

d. Fixity
    i. Standard (partial, on ingest): http://tools.ietf.org/html/rfc3230#section-4.3.2
    ii. key summary type created updated due assignee reporter priority status resolutic

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

e. Authorization
    i. Standard: WebAC
    ii. type key summary assignee reporter priority status resolution created updated du

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

2. Formalize the core Fedora Service Provider Interfaces (SPIs)

    a. Messaging SPI
        i. Defining the interface that a Fedora repository implementation should implement to publish repository events
3. Runtime configurability
    a. Enable the update of configuration settings at runtime, e.g. changing hostname published in repository events
    b. Enable pluggability of extension modules, e.g. adding an OAI-PMH module at runtime
4. Performance and Scale
    a. Establish metrics for repository limits, including:
        i. number of resources
        ii. number of bytes
        iii. See: Performance and Scalability Test Plans

        iv. type key summary assignee reporter priority status resolution created updated du

        Unable to locate Jira server for this macro. It may be due to Application Link configuration.

    b. Establish guidelines for storage options based on usage patterns

Note: Items 1 and 2 define priorities related to "Fedora as a specification", whereas Items 3 and 4 relate to "Fedora as an implementation".

2015-2016 Technical Roadmap

1. Formalize the core Fedora services Application Programming Interface (API)
   This priority is to clearly define the core services that Fedora promises as a standards-based RESTful API, accompany this API with any necessary domain-specific ontologies, and provide a Technology Compatibility Kit (TCK) for each service.

type key summary assignee reporter priority status resolution created updated due

Unable to locate Jira server for this macro. It may be due to Application Link configuration.

The Fedora services are:

    a. Create/Read/Update/Delete on repository resources
        i. Standard: Linked Data Platform
        ii. Include Import and Export of RDF, and option for RDF serialization to disk

        iii. type key summary assignee reporter priority status resolution created updated du

        Unable to locate Jira server for this macro. It may be due to Application Link configuration.

    b. Versioning
        i. Standard (partial, only retrieval): Memento

        ii. type key summary assignee reporter priority status resolution created updated du

        Unable to locate Jira server for this macro. It may be due to Application Link configuration.

    c. Transactions
        i. Standard: TBD
    d. Fixity

i. Standard (partial, on ingest): http://tools.ietf.org/html/rfc3230#section-4.3.2

e. Authorization
   i. Standard: WebAC

ii. type key summary assignee reporter priority status resolution created updated du

> Unable to locate Jira server for this macro. It may be due to Application Link configuration.

2. Formalize the core Fedora Service Provider Interfaces (SPIs)
   a. Eventing SPI
      i. Defining the interface that a Fedora repository implementation should implement to publish repository events
3. Runtime configurability
   a. Enable the update of configuration settings at runtime, e.g. changing hostname published in repository events
   b. Enable pluggability of extension modules, e.g. adding an OAI-PMH module at runtime
4. Performance and Scale
   a. Establish metrics for repository limits, including:
      i. number of resources
      ii. number of bytes
      iii. See: Performance and Scalability Test Plans

iv. type key summary assignee reporter priority status resolution created updated du

> Unable to locate Jira server for this macro. It may be due to Application Link configuration.

   b. Establish guidelines for storage options based on usage patterns

Note: Items 1 and 2 define priorities related to "Fedora as a specification", whereas Items 3 and 4 relate to "Fedora as a reference implementation".

Previous Technical Roadmap Items

Currently Supported Features	Design	Core	Non-core	4.0	Use Cases
AuthN/Z	design		x	✅	• Hydra Authorization Use Case • Yale University - Fedora managing access conditions • University of Virginia - Repository generated-mediated derivatives
Backup	design	x		✅	• Backups / Disaster Recovery
Clustering		x		⚠️	• Consistent deployment • REST-API support against master node
Content Modeling - Structural		x		⚠️	**Content by label** There is no content with the specified labels
Managed External Datastreams			x	✅	• Ingesting large files into the repository
Store/Deliver Large Files	design	x		✅	• Ingesting large files into the repository • Research Data
Search	design		x	✅	• University of Virginia - Applications can be easily built to work against fedora

Transactions		x		✅	• Islandora CLAW Transaction Use Case • University of Virginia - Applications can be easily built to work against fedora
Triplestore	design		x	✅	• University of Virginia - Live querying of object graph • Using the resources of the Semantic Web to describe repository contents
Versioning		x		✅	• Structure - Automated Retention of All changes to Items • Structure - Manual Edit of Existing Items • Structure - Manual Creation of "New Editions" of an Item • Structure - Generated provenance for all added bitstreams
Non-Functional: Easy Deployment				✅	
Non-Functional: Performance - Single-node				⚠️	• Improve Performance • University of North Carolina at Chapel Hill - High Volume of Concurrent Ingests
Post-4.0 Priority 1 Features	**Design**	**Core**	**Non-core**	**4.0**	**Use Cases**
3 to 4 Upgrade	design		x		**Content by label** There is no content with the specified labels
Audit Service	design	x			• University of New South Wales - Viewing-exporting usage statistics about fedora records • More Functional AUDIT Datastream • Reporting Functionality
Managed External Datastreams - Indexing			x		
Asynchronous storage API	design	x			• Support for Hierarchical Storage Management-like systems • Scalability for Asynchronous Ingest, Access and Audit • Leverage Cloud Storage • Repository supports Amazon Glacier, or other asynchronous storage services • Hybrid Use Case
Asynchronous storage Implementation		x			• Support for Hierarchical Storage Management-like systems • Scalability for Asynchronous Ingest, Access and Audit • Leverage Cloud Storage • Repository supports Amazon Glacier, or other asynchronous storage services • Hybrid Use Case
LDP-Paging		x			
Web Access Control			x		
API Partitioning		x			
Post-4.0 Priority 2 Features	**Design**	**Core**	**Non-core**	**4.0**	
Batch Operations		x			• Yale University - Fedora managing access conditions • Hybrid Use Case (Metadata Services)
CMIS			x		

Content Modeling - Services and Validation					• Use case: transform application workflow • Art Institute of Chicago Use Case - Structural Validation • Modeling content in concert with the wider world • Art Institute of Chicago Use Case - Structural Validation - Properties
Disseminator-like Functionality			x		• University of Virginia - Repository-level metadata transformations-mapping • University of Virginia - Repository generated-mediated derivatives • Objects can be associated with a descriptive metadata service
Human-readable Filesystem Storage			x		
Metrics		x			• University of New South Wales - Viewing-exporting usage statistics about fedora records • More Functional AUDIT Datastream • Reporting Functionality
Multi-tenancy		x			• Multi-tenancy Single Fedora Instance
OAI-PMH	design		x		
ORCID Support			x		
Policy-driven Storage	design	x			• Policy-controlled storage • Hybrid Use Case
Relationships API		x			
Self-healing Storage			x		
WebDAV			x		
Non-Functional: Performance - Clustered					• Mass Object Creation • Repository subsystems can scale horizontally • Scale for High Volume Read Access • University of North Carolina at Chapel Hill - High Volume of Concurrent Ingests
Previously Un-prioritized Features	**Design**	**Core**	**Non-core**	**4.0**	**Use Cases**
Admin UI			x	✅	• Tufts University - Reasonable curators administrative interface
Content API		x		✅	• Yale University - Fedora managing access conditions • University of Virginia - Applications can be easily built to work against fedora
Identifiers		x		✅	• Hybrid Use Case (Metadata Services) • UUID's: Object Mobility and Merging Stores
Large-Scale Content		x		⚠️	• Archive of digitised newspapers