Introduction and Feature Tour

David Wilcox, DuraSpace
Learning Outcomes

- Understand the purpose of a Fedora repository
- Learn what Fedora can do for you
- Understand the key capabilities of the software
Introduction to Fedora 4
What is a Fedora Repository?

- Secure software that stores, preserves, and provides access to digital materials
- Supports complex semantic relationships between objects both within and outside the repository
- Supports millions of objects, both large and small
- Capable of interoperating with other applications and services
Exposing and Connecting Content

- Flexible, extensible content modeling
- Atomic resources with semantic connections using standard ontologies
- RDF-based metadata using Linked Data
- RESTful API with native RDF response format
Fedora 4 Project Goals

- Improved performance
- Flexible storage options
- Research data management
- Linked open data support
- Improved platform for developers
# New Vocabulary

<table>
<thead>
<tr>
<th>Fedora 3</th>
<th>Fedora 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects and datastreams</td>
<td>Resources</td>
</tr>
<tr>
<td>Object</td>
<td>Container</td>
</tr>
<tr>
<td>Datastream</td>
<td>Binary</td>
</tr>
</tbody>
</table>
Data Modeling
Resources

- Both containers and binaries are resources.
- Container resources can have both containers and binaries as children.
- The tree structure allows for inheritance of things like security policies.
Properties

- Resources have a number of properties, expressed as RDF triples
  - Name-value pairs; translated to RDF on REST-API responses
- Properties can be RDF literals or URIs
- Any number of RDF namespaces can be defined and used
Content Models

- Content can be modeled using properties and types
- Cross-community design has produced PCDM
  - Portland Common Data Model
- PCDM combines common ontology with LDP interaction model
Linked Data

- Fedora 4 conforms to the LDP 1.0 recommendation
- Metadata can be represented as RDF triples that point to resources inside and outside the repository
- Many possibilities for exposing, importing, sharing resources with the broader web
Core Features
Standards

● Focus on existing standards
● Fewer customizations to maintain
● Opportunities to participate in related communities
Core Features and Standards

- CRUD - LDP
- Versioning - Memento?
- Authorization - WebAC?
- Transactions
- Fixity
- Import/Export - RDF export?
Versioning

- Versions can be created on resources with an API call
- A previous version can be restored via the REST-API
Authorization

• The authorization framework provides a plug-in point within the repository that calls out to an optional authorization enforcement module

• Currently, three authorization implementations exist: No-op, Role-based and XACML
Role-based Authorization

- Role-based authorization compares the user's role(s) with an Access Control List (ACL) defined on a Fedora resource.
- ACLs can be inherited; if a given resource does not have an associated ACL, Fedora will examine parent resources until it finds one.
XACML Authorization

- A default policy must be defined for the repository, and each resource can override the default with another policy.
- An XACML policy referenced by a resource will also apply to all the resource's children, unless they define their own XACML policies that override the parent policy.
Transactions

- Multiple actions can be bundled together into a single repository event (transaction)
- Transactions can be rolled back or committed
- Can be used to maintain consistency
Fixity

- Over time, digital objects can become corrupt
- Fixity checks help preserve digital objects by verifying their integrity
- On ingest, Fedora can verify a user-provided checksum against the calculated value
- A checksum can be recalculated and compared at any time via a REST-API request
Export and Import

- A specific Fedora container, its child containers, and associated binaries can be exported.
- Exported containers can be serialized in a standard RDF format.
- An exported container or hierarchy of containers can be imported at any time.
Backup and Restore

- A full backup can be performed at any time
- A full restore from a repository backup can be performed at any time
Non-Core Features
Two Feature Types

- Optional, pluggable components
  - Separate projects that can interact with Fedora 4 using a common pattern
- External components
  - Consume and act off repository messages
External Component Integrations

- Leverages the well-supported Apache Camel project
  - Camel is middleware for integration with external systems
  - Can handle any asynchronous, event-driven workflow
External - Indexing

- Index repository content for search
- Content can be assigned the rdf:type property "Indexable" to filter from non-indexable content
- Solr has been tested
External - Triplestore

- An external triplestore can be used to index the RDF triples of Fedora resources
- Any triplestore that supports SPARQL-update can be plugged in
  - Fuseki and Sesame have been tested
External - Audit Service

- Maintains a history of events for each repository resource
- Both internal repository events and events from external sources can be recorded
- Uses the existing event system and an external triplestore
Pluggable - OAI Provider

- fcrepo4-oaiprovider implements Open Archives Protocol Version 2.0 using Fedora 4 as the backend
- Exposes an endpoint which accepts OAI conforming HTTP requests
- Supports oai_dc out of the box, but users are able to add their own metadata format definitions to oai.xml
Pluggable - SWORD Server

- SWORD is a lightweight protocol for depositing content from one location to another
- fcrepo4-swordserver implements 2.0 AtomPub Profile, using Fedora 4 as the backend
- SWORD v2 includes AtomPub CRUD operations
Performance
A number of scalability tests have been run:

- Uploaded a 1 TB file via REST API
- 16 million objects via federation
- 10 million objects via REST API
Multiple actions can be bundled together into a single repository event (transaction)

Transactions offer performance benefits by cutting down on the number of times data is written to the repository filesystem (which tends to be the slowest action)
Clustering

- Two or more Fedora instances can be configured to work together in a cluster.
- Fedora 4 currently supports clustering for high-availability use cases.
- A load balancer can be setup in front of two or more Fedora instances to evenly distribute read requests across each instance.
Further Reading

● Fedora 4 Wiki
  ○ https://wiki.duraspace.org/display/FF/Fedora+Four+Prospectus

● Fedora 4 Documentation
  ○ https://wiki.duraspace.org/display/FEDORA4x/Fedora+4.x+Documentation