Training - Migrating from Fedora 3 to Fedora 4

These training archives may be out of date, but have been retained and kept available for the community's benefit in reviewing previous sessions.

Current training documentation can be found here: Training

Learning Outcomes

- Understand the main differences between Fedora 3 and Fedora 4
- Understand the considerations necessary for migrating from Fedora 3 to Fedora 4
- Explore new possibilities for enhancing data in Fedora 4

Differences Between Fedora 3 and Fedora 4

XML Objects vs. Nodes

Fedora 3	Fedora 4
FOXML objects	ModeShape nodes (still referred to as objects)
Inline and managed XML	RDF properties*

**XML Datastreams are still supported*

Flat Filesystem vs. Hierarchy

Fedora 3	Fedora 4
Objects and Datastreams stored in flat directories	Objects and datastreams stored in a hierarchical filesystem

PIDs vs. Path

Fedora 3	Fedora 4
Objects have Persistent Identifiers (PIDs)	Objects have a path (including a UUID) based on their location in the file system hierarchy
	Objects can also have other identifiers (DOIs, Handles, PIDs, etc.)

Migration Considerations

To Ingest or Federate?

- In addition to normal ingest, Fedora 4 supports federation over content in a existing file system.
- Federating over Fedora 3 content is possible, but the connector would need to be updated and modified to suit a particular use case.
- Federation also does not provide any opportunities for data enhancement, which is an important consideration.

Security

- What kind of security does your Fedora 3 repository use?
- Many Fedora 3 repositories use XACML security; this is supported in Fedora 4.
- However, it would be prudent to test your specific XACML policies within the context of Fedora 4.
    - While XACML in general is supported, broad testing of existing policies is recommended.

Versions

- Does your Fedora 3 repository use versioning?
- What versions do you want to preserve? Object or datastream level?
- How should version dates be handled? Will you use the system modified date, or a special date property?

Content Models

- How are content models used in your Fedora 3 repository?
    - Do they have any logic built into them, or is that handled at a higher application level (e.g. Islandora, Hydra)?
- Are there opportunities for building common content models that could be shared between Fedora implementations?
    - E.g. Islandora content models that can also be recognized by Hydra.

Disseminators

- Does your Fedora 3 repository make use of disseminators?
- What are they used for? XSLT transformations? Something else?
- How can we support the existing disseminator use cases in Fedora 4 without re-creating disseminators?

Other Considerations

- Have we left anything out?
- What else needs to be considered when planning a migration?

Enhancements

Taking Advantage of Properties

- Converting Inline XML and/or XML Datastreams (e.g. RELS-EXT, RELS-INT) to RDF properties.
    - Inline XML is no longer supported.
- Lightweight compared to XML.
- New possibilities for complex queries that extend beyond the limits of the repository.
    - Linked data relationships can be exposed via a standardized REST-API
    - Web applications can take advantaged of these standardized representations.
    - Data can be shared and manipulated in new and interesting ways.

Enhancing Your Metadata

- XML metadata datastreams are still supported, but there are new opportunities to explore!
- XML metadata can be converted into RDF metadata using an RDF-based schema.
- RDF metadata is easier to query and share.
- Take advantage of linked data by pointing to authority URIs.