Migrating from Fedora 3 to 4
Topics

● Differences between Fedora 3 and 4
● Migration overview
● Data migration planning
● Functionality migration planning
● Data migration with “migration-utils”
Differences Between Fedora 3 and Fedora 4
Conceptual Models of Repository Resources

Fedora 3
- Content Model Architecture
- Objects: Collect bytestreams & properties
- Datastreams: Bytestreams in context of an object, with some properties

Fedora 4
- Linked Data Platform
- LDP RDF resources (objects & containers)
- LDP non-RDF binaries (& description)
What About **PCDM**?
Organization of Repository Entities

Fedora 3: Flat

- Objects and datastreams at the top level
- No inherent tree structure

Fedora 4: Hierarchy

- Containers and binaries in a hierarchy
- All resources descend from a root resource
Storage of Repository Data

Fedora 3: Akubra
- Objects directory and datastreams directory
- Both objects and datastreams are in a PairTree

Fedora 4: Infinispan & other MODEism
- Containers in a database (e.g. LevelDB)
- Datastreams in a PairTree directory
Identification of Repository Resources

Fedora 3: PID

- Objects have Persistent Identifiers (PID)
- Uniform structure
- An object’s PID can never be altered

Fedora 4: Path

- Resources have a repository path
- This can be user-defined or generated via an ID-minter
## How Do These Concepts Correlate?

<table>
<thead>
<tr>
<th>Fedora 3/CMA</th>
<th>Fedora 4/LDP</th>
<th>PCDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>RDFSource/Container</td>
<td>AdminSet/Collection/Object</td>
</tr>
<tr>
<td>Datastream</td>
<td>NonRDFSource</td>
<td>File</td>
</tr>
<tr>
<td>PID</td>
<td>Path</td>
<td>“id”</td>
</tr>
<tr>
<td>Akubra (local)</td>
<td>Infinispan (clusterable)</td>
<td>n/a</td>
</tr>
</tbody>
</table>
Data Mapping
### fcrepo3 Object properties to fcrepo4

<table>
<thead>
<tr>
<th>fcrepo 3</th>
<th>fcrepo4</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID</td>
<td>fedora3model:PID†</td>
<td>yul:328697</td>
</tr>
<tr>
<td>state</td>
<td>fedoraaccess:objState</td>
<td>Active</td>
</tr>
<tr>
<td>label</td>
<td>fedora3model:label†</td>
<td>Elvis Presley</td>
</tr>
<tr>
<td>createDate</td>
<td>premis:hasDateCreatedByApplication</td>
<td>2015-03-16T20:11:06.683Z</td>
</tr>
<tr>
<td>lastModifiedDate</td>
<td>metadataModification</td>
<td>2015-03-16T20:11:06.683Z</td>
</tr>
<tr>
<td>ownerId</td>
<td>fedora3model:ownerId†</td>
<td>nruest</td>
</tr>
<tr>
<td>fcrepo3</td>
<td>fcrepo4</td>
<td>Example</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>DSID</td>
<td>dcterms:identifier</td>
<td>OBJ</td>
</tr>
<tr>
<td>Label</td>
<td>dcterms:title†</td>
<td>ASC19109.tif</td>
</tr>
<tr>
<td>MIME Type</td>
<td>ebucore:hasMimeType†</td>
<td>image/tiff</td>
</tr>
<tr>
<td>State</td>
<td>fedoraaccess:objState</td>
<td>Active</td>
</tr>
<tr>
<td>Created</td>
<td>premis:hasDateCreatedByApplication</td>
<td>2015-03-16T20:11:06.683Z</td>
</tr>
<tr>
<td>Versionable</td>
<td>fedora:hasVersions†</td>
<td>true</td>
</tr>
<tr>
<td>Format URI</td>
<td>premis:formatDesignation†</td>
<td>info:pronom/fmt/156</td>
</tr>
<tr>
<td>Alternate IDs</td>
<td>dcterms:identifier†</td>
<td></td>
</tr>
<tr>
<td>Access URL</td>
<td>dcterms:identifier†</td>
<td></td>
</tr>
</tbody>
</table>
| Checksum         | cryptofunc:hashalgorithm†                   | cryptofunc:sha1       
|                  |                                              | "c91342b705b15cb4f6ac5362cc6a47d9425" |
Migration Concerns

Data Migration
  ● Metadata format migration

Functionality Migration
  ● Access controls
  ● Service migration
  ● Integrations
Data Migration Planning: Purpose

Access vs Preservation
Data Migration Planning: Preservation Needs

Is any loss/ambiguities in metadata tolerable?

Is any transformation of metadata serialization tolerable?
Functionality Migration Planning: Purpose

What behaviors of Fedora 3 would you like to carry over in Fedora 4?
Functionality Migration Planning: Access Controls

How complex are your access control scenarios?

- Broad policies
- Collection policies may impact F4 organization
- Per object policies
- Granularity may impact migration strategies (XML to RDF)
Functionality Migration Planning: Service Migrations

What is the nature of your service deployments?

- Metadata transformations?
- Image derivatives?
- Something else?
Functionality Migration Planning: Integration and Access Needs

What content is used by external systems or has been advertised to users?

What expectations for continued and consistent access are there?
Using migration-utils

a starting point
About migration-utils

- Collaboratively developed
- Open source
- Java
- Command line
- Front-end agnostic

https://github.com/fcrepo4-exts/migration-utils
Getting started

Make sure you have java 8

Download the utility distribution
Getting started (curl)

curl -L http://git.io/vZFtK -o "mu.zip"

unzip mu.zip
Getting started (wget)

wget -O mu.zip http://git.io/vZFtK

unzip mu.zip
Getting started

README at https://github.com/fcrepo4-exts/migration-utils

Background work

- Determine your migration source FOXML
  - Akubra
  - Legacy fs
  - Exported
  - Fedora 2

- Determine your destination location
  - http://localhost:8080/fcrepo/rest
Update sample configuration

migration-utils/conf/fedora3-akubra.xml

This configuration is the place to start if your scenario involves migrating data files from a relatively recent Fedora 3 installation.

There are other sample files that serve as starting points for different migration scenarios.
Update the sample configuration

Update the “fedora4Url” to point to your vagrant instance: http://localhost:8080/fcrepo/rest
Peruse the other likely options

All of the likely changes are clearly marked in comments with “CHANGE ME IF ...”

- Ingest limit
- External and redirect datastream handling
- Fedora 3 objects and datastreams directory
Run the migration

```
java -jar migration-utils-4.3.1-SNAPSHOT-driver.jar
conf/fedora3-akubra.xml
```
View the results