Extended Services

David Wilcox, DuraSpace
@d_wilcox
What is Messaging

Fundamental concepts
What is Messaging?

A mechanism for notifying interested parties that something has happened

- “Mechanism”: Framework for delivering messages (e.g. ActiveMQ)
- “Interested parties”: External software components/services that listen for events
- “Something”: A resource in the repository has changed
- “Has happened”: Messages are asynchronous; describe events that have already occurred
Key characteristics

- From repository’s perspective “fire and forget”
- Messaging framework responsible for delivery guarantees (choice and configuration of messaging software)
  - Durability of messages
  - Timeliness of message
  - Ordering of messages
  - Enqueueing messages
- Clients operate asynchronously
  - Message arrives some time after event occurred
  - Can operate at their own pace without affecting other clients, or the repository
Messages

• Headers + body (much like an HTTP response)
• Body: unconstrained. Fedora uses JSON-LD messages.
  ○ Know what kind/format of messages you’re going to get before you subscribe
• Anatomy of a message from Fedora:
  ○ Resource URI
  ○ rdf:type of resource
  ○ Parent resource URI
  ○ Type of event (C, U, D)
  ○ Time event occurred
  ○ User
• Notably and intentionally absent: Content of resource
When does Fedora 4 emit messages?

When Events happen that are related to *durable* changes to your resources in Fedora 4 (CUD)

- CREATE
- UPDATE
- DELETE
What does not trigger a message?

- **Batch atomic operations**
  - Rollbacks (they were never persisted in storage)
  - Commits (each CRUD operation that results from the commit triggers a message, but not the commit itself)

- **CLIENT/REST API errors**
  - It’s not a log system, it’s an async communication system!

- **Read-only service invocations (e.g. checking fixity)**
  - We’ll explore fixity later in this presentation
Triplestore Indexing
fcrepo-indexing-triplestore
Hands-On: Indexing in triplestore

http://localhost:8080/fuseki
Hands-On: Indexing in triplestore

```sparql
select * where {
}
```
```sparql
select * where {
}
```

<table>
<thead>
<tr>
<th>p</th>
<th>o</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdfs:type</td>
<td><a href="http://fedora.info/definitions/v4/repository#Container">http://fedora.info/definitions/v4/repository#Container</a></td>
</tr>
<tr>
<td>rdfs:type</td>
<td><a href="http://fedora.info/definitions/v4/repository#Resource">http://fedora.info/definitions/v4/repository#Resource</a></td>
</tr>
<tr>
<td>rdfs:type</td>
<td>ldp:Container</td>
</tr>
<tr>
<td>rdfs:type</td>
<td>ldp:RDFSource</td>
</tr>
<tr>
<td><a href="http://fedora.info/definitions/v4/repository#writable">http://fedora.info/definitions/v4/repository#writable</a></td>
<td>true^^xsd:boolean</td>
</tr>
<tr>
<td><a href="http://fedora.info/definitions/v4/repository#lastModified">http://fedora.info/definitions/v4/repository#lastModified</a></td>
<td>2017-10-06T18:28:37.008Z^^xsd:dateTime</td>
</tr>
<tr>
<td><a href="http://fedora.info/definitions/v4/repository#created">http://fedora.info/definitions/v4/repository#created</a></td>
<td>2017-10-06T18:28:37.008Z^^xsd:dateTime</td>
</tr>
<tr>
<td><a href="http://fedora.info/definitions/v4/repository#createdBy">http://fedora.info/definitions/v4/repository#createdBy</a></td>
<td>bypassAdmin</td>
</tr>
</tbody>
</table>
Solr Indexing
Hands-On: Indexing in Solr

http://localhost:8080/solr
Hands-On: Indexing in Solr
{
  "responseHeader": {
    "status": 0,
    "QTime": 0,
    "params": {
      "q": "",
      "wt": "json",
      "indent": "true",
      "_" : "1478022760663"
    }
  },
  "response": {
    "numFound": 5,
    "start": 0,
    "docs": [
      {
        "id": "http://localhost:8888/fcrepo/rest/pcdm-object/files",
        "hasParent": {
          "http://localhost:8888/fcrepo/rest/pcdm-object"
        },
        "created": "1478823569999",
        "lastModified": "1478823569999",
        "lastModifiedBy": {
          "FedoraAdmin"
        },
        "version": 1549831906657816000,
        "id": "http://localhost:8888/fcrepo/rest/pcdm-object/files/0e/cc/ec/0e35ec-8767-44da-bafe-5acdc4d60d01",
        "version": 15498319067838235880
      },
      ...
    ]
  }
}
Preservation Services
Preservation Is No Single Action

Fedora features that support digital preservation:

- Persistence
- Fixity
- Versioning
- Audit
- Import / Export

http://fedorarepository.org/fedora-and-digital-preservation
Why Import/Export?

Transfer repository resources into/out of preservation systems

- Standardized serialization of resources
- BagIt bags
- Future-proofing repository resources

Transfer repository resources between Fedora installations

- Repository version upgrades
- Repository implementation migrations
- Disaster recovery
Import / Export Architecture

File System
- Exported Objects
- Exported Binaries

Import/Export Utility

Verification Utility

Fedora
- Objects
- Binaries
Export from Repository

File System
- Exported Objects
- Exported Binaries

Import/Export Utility

Verification Utility

Fedora
- Objects
- Binaries
Import to Repository

File System
- Exported Objects
- Exported Binaries

Import/Export Utility

Verification Utility

Fedora
- Objects
- Binaries
Hands-On: Set-Up

Ensure Fedora is running:

http://localhost:8080/fcrepo/rest/

Ensure data is loaded:

- Object(s)
- Binaries
- Optional: External binaries

Download Import/Export Utility
Tool overview and usage
Tool Usage

Running the Import/Export utility:

$ java -jar fcrepo-import-export-0.2.0.jar [options]

...Place the fcrepo-import-export-0.2.0.jar into the VM directory, and give it a try (no options)
Options: Import/Export Utility

Running Import/Export Utility from command line arguments


-a,--auditLog       Enable audit log creation, disabled by default
-b,--binaries      When present this flag indicates that binaries should be imported/exported.
-d,--dir <dir>      The directory to export repo to or import the repo from.
-g,--bag-profile <profile>      Export and import BagIt bags using profile [default|aptrust]
-G,--bag-config <path>      Path to the bag config file
-h,--help          Print these options
-i,--inbound       When present this flag indicates that inbound references should be exported.

...
Options: Import/Export Utility --- cont 2

-L,--legacyMode
-1,--rdfLang <rdfLang>
-m,--mode <mode>
-M,--map <map>
-p,--predicates <predicates>
-r,--resource <resource>
-t,--overwriteTombstones
-u,--user <user>
-V,--versions

When importing, omit certain server-managed-triples that aren't modifiable in old versions of fedora.

RDF language (default: text/turtle)

Mode: [import|export]

Old and new base URIs, separated by comma, to map URIs when importing

Comma-separated list of predicates to define resource containment

Resource (URI) to import/export

When importing, overwrite "tombstones" left behind after resources were deleted.

username:password for fedora basic authentication

When exporting, include versions of resources and binaries.
Options: Import/Export Utility --- cont 3

-\texttt{w,--writeConfig} <writeConfig> When present this flag indicates that a sample config should be written at the specified filename.

-\texttt{x,--external} When present this flag indicates that external content should be exported.

--- or ---

Running Import/Export Utility from configuration file
usage: java -jar import-export-driver.jar -c <config> [-u <user>]
-c,--config <config> Path to config file
-u,--user <user> username:password for fedora basic authentication
Export: Basic

$ java -jar fcrepo-import-export-0.2.0.jar \ 
--user fedoraAdmin:secret3 \ 
--mode export \ 
--resource http://localhost:8080/fcrepo/rest \ 
--dir data-dir \ 
--binaries
Export: Basic

$ java -jar fcrepo-import-export-0.2.0.jar \
-u fedoraAdmin:secret3 \
-m export \
-r http://localhost:8080/fcrepo/rest \
-d data-dir \
-b
File System View

data/
├── fcrepo
│   ├── rest
│   │   ├── basic
│   │   │   └── collection.ttl
│   │   │   └── images
│   │   │       ├── access
│   │   │       │   └── fcr%3Ametadata.ttl
│   │   │       └── access.binary
│   │   │           └── images.ttl
│   │   └── basic.ttl
│   ├── Book.ttl
│   ├── master
│   │   └── fcr%3Ametadata.ttl
│   └── master.external
│       └── rest.ttl
Clear the repository

$ vagrant halt

$ vagrant destroy

$ vagrant up

Verify empty repository:

http://localhost:8080/fcrepo/rest/
Import: Basic

$ java -jar fcrepo-import-export-0.2.0.jar \ 
-u fedoraAdmin:secret3 \ 
-m import \ 
-r http://localhost:8080/fcrepo/rest \ 
-d data-dir \ 
-b \ 
--legacyMode
Other options

-a,--auditLog
Enable audit log creation, disabled by default

-w,--writeConfig <writeConfig>
When present this flag indicates that a sample config should be written at the specified filename.

-u,--user <user>
username:password for fedora basic authentication

-V,--versions
When exporting, include versions of resources and binaries.

-l,--rdfLang <rdfLang>
RDF language (default: text/turtle)
Export: Default

BookA

BookB

BookC

SeriesX

Page0

Page1

OCR

ldp:contains

ldp:contains

pcdm:hasMember
Export Inbound References

-\texttt{i,--inbound}

When present this flag indicates that inbound references should be exported.
Customize membership predicate(s)

-p, --predicates <predicates>
Comma-separated list of predicates to define resource containment

- Default:
  - http://www.w3.org/ns/ldp#contains

- Other options:
  - http://pcdm.org/models#hasMember
  - http://www.openarchives.org/ore/terms/proxyFor
  - etc.
--predicates

BookA

BookB

BookC

SeriesX

Page0

Page1

OCR

ldp:contains

ldp:contains

ldp:contains

pcdm:hasMember

-p http://pcdm.org/models#hasMember
--predicates

BookA ldp:contains Page0
BookB ldp:contains Page0
BookC ldp:contains Page1
SeriesX pcdm:hasMember OCR

-p http://pcdm.org/models#hasMember
Importing and Exporting Bags

- Export Bags conforming to the Library of Congress BagIt specification
- Very similar to normal export format, with some additions:
  - Data moved to data subdirectory
  - Metadata included in tag files (bagit.txt, bag-info.txt, etc.)
  - Checksums of all data and tag files (tagmanifest-sha1.txt, etc.)
- Details:
  - https://github.com/fcrepo4-labs/fcrepo-import-export#running-the-importexport-utility-with-a-bagit-support
Bag Profiles

-g, --bag-profile <profile>
Export and import BagIt bags using profile [default|aptrust]

- Customize checksums generated, required metadata, etc.
  - https://github.com/ruebot/bagit-profiles
  - default
  - aptrust
  - metaarchive
  - perseids
- Custom profile for your own requirements, or to customize builtins
Bag Metadata

-G, --bag-config <path>
Path to the bag config file

- Varies by profile, some have more requirements than others
- Can populate bag-info.txt, separate tag files
- Minimal metadata (create new ‘metadata.yml’ file):

```yaml
### start metadata.yml ###
bag-info.txt:
  Source-Organization: Example
  Contact-Name: Fedo Raadmin
### end metadata.yml ###
```
Export Bags

$ java -jar fcrepo-import-export-0.2.0.jar \
-u fedoraAdmin:secret3 \
-m export \
-d data-dir3 \
-r http://localhost:8080/fcrepo/rest \
-b \
-g default \
-G metadata.yml
Verification Tool
The Import or export is complete. Now what?

Tool designed to verify that the import or export succeeded in including all the resource data correctly.

- Verify data export/imported correctly
- All triples accounted for
- Detect any corruption of binaries or triples

Creates:

- Detailed log
- CSV file with resource & validation information
Technical Details

For **import** the verification tool will walk the originating directory structure and verify that the data is in the Fedora 4 server correctly.

For **export** the verification tool will walk the Fedora 4 server and verify that the corresponding data is the same on disk.

Reads same configuration file that the import/export tool creates.

Python3 with RDFLib to work with the data

https://github.com/fcrepo4-labs/fcrepo-import-export-verify
What did we learn?

- Import / Export Service

Questions?