Export Format

Camel serializer

The camel serializer exists as a proof-of-concept serialization of resources within the repository. The intention is to have an eventually consistent copy of the repository content. In working to build lossless import/export functionality, the serialization format is used as a starting point.

Using the Camel Serializer

The Camel Serializer is automatically installed With the Fedora 4.6.0 release of the fcrepo-vagrant test application. Creating a sample of the output from the serializer can be done with the following steps:

- git clone https://github.com/fcrepo4-exts/fcrepo4-vagrant.git
- cd fcrepo4-vagrant
- vagrant up
- vagrant ssh
- (optional) sudo vi /opt/karaf/etc/org.fcrepo.camel.serialization.cfg
  change line that says:
  serialization.includeBinaries=false
  to:
  serialization.includeBinaries=true
- In a web browser, go to http://localhost:8080/fcrepo/rest
- create a new container, and add a binary to that container
- cd /tmp/descriptions, to see the metadata that was saved by the serializer
- cd /tmp/binaries to see the binaries that were saved (assuming you set includeBinaries=true

Only content that is added or changed while the Serializer is running will be copied to /tmp/descriptions and /tmp/binaries.

An Example

Following the instructions above, I created a new empty 4.6.0 Fedora repository.
I then created a new container (via the web interface) at http://localhost:8080/fcrepo/rest/album

Inside that container I uploaded a binary file, without specifying an identifier (i.e. let Fedora auto-generate an identifier). This resulted in a binary file at http://localhost:8080/fcrepo/rest/album/ea/50/12/93/ea501293-64bf-430e-bf97-abcd64fda0c4 (the identifier will be different for you if you reproduce these steps).

Next, I looked in /tmp/descriptions and /tmp/binaries and found output that was generated by the Camel Serializer

Sample Output

The metadata for the album container I created at **http://localhost:8080/fcrepo/rest/album** was saved to a turtle file at **/tmp/descriptions/albums.ttl**.

The metadata for the jpg binary file I uploaded to **http://localhost:8080/fcrepo/rest/album/ea/50/12/93/ea501293-64bf-430e-bf97-abcd64fda0c4** was saved to a turtle file at **/tmp/descriptions/albums/ea/50/12/93/ea501293-64bf-430e-bf97-abcd64fda0c4.ttl**.

The jpg file was saved as a binary at **/tmp/binaries/albums/ea/50/12/93/ea501293-64bf-430e-bf97-abcd64fda0c4** .

I manually confirmed that the md5sum of this binary in /tmp/binaries matched the original files checksum.

There are additional ttl files in /tmp/descriptions/fedora:system that contain Fedora generated triples, as well as 2 binaries in /tmp/binaries/fedora:system, which I ignored as they are not part of the Resource I am interested in serializing in this test.

```
vagrant@fedora4:/tmp$ tree descriptions
descriptions/
 albums
    ea
        50
            12
                93
                   ea501293-64bf-430e-bf97-abcd64fda0c4.ttl
 albums.ttl
 fedora:system
     fedora:transform
        fedora:ldpath
           default
              fedora:Resource.ttl
           default.ttl
           deluxe
              fedora:Resource.ttl
           deluxe.ttl
     fedora:transform.ttl

vagrant@fedora4:/tmp$ tree binaries/
binaries/
 albums
    ea
        50
            12
                93
                   ea501293-64bf-430e-bf97-abcd64fda0c4
 fedora:system
     fedora:transform
         fedora:ldpath
            default
               fedora:Resource
            deluxe
                fedora:Resource

vagrant@fedora4:/tmp/descriptions$ cat albums.ttl
@prefix premis: <http://www.loc.gov/premis/rdf/v1#> .
@prefix image: <http://www.modeshape.org/images/1.0> .
@prefix sv: <http://www.jcp.org/jcr/sv/1.0> .
@prefix test: <info:fedora/test/> .
@prefix nt: <http://www.jcp.org/jcr/nt/1.0> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsi: <http://www.w3.org/2001/XMLSchema-instance> .
@prefix mode: <http://www.modeshape.org/1.0> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix fedora: <http://fedora.info/definitions/v4/repository#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix audit: <http://fedora.info/definitions/v4/audit#> .
@prefix jcr: <http://www.jcp.org/jcr/1.0> .
@prefix ebucore: <http://www.ebu.ch/metadata/ontologies/ebucore/ebucore#> .
@prefix ldp: <http://www.w3.org/ns/ldp#> .
@prefix xs: <http://www.w3.org/2001/XMLSchema> .
@prefix fedoraconfig: <http://fedora.info/definitions/v4/config#> .
@prefix mix: <http://www.jcp.org/jcr/mix/1.0> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .


<http://localhost:8080/fcrepo/rest/albums> a fedora:Container , fedora:Resource ;
    fedora:lastModifiedBy "bypassAdmin"^^<http://www.w3.org/2001/XMLSchema#string> ;
    fedora:createdBy "bypassAdmin"^^<http://www.w3.org/2001/XMLSchema#string> ;
    fedora:created "2016-09-06T15:15:55.666Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> ;
    fedora:lastModified "2016-09-06T15:16:34.844Z"^^<http://www.w3.org/2001/XMLSchema#dateTime> ;
    a ldp:RDFSource , ldp:Container ;
    fedora:writable "true"^^<http://www.w3.org/2001/XMLSchema#boolean> ;
    fedora:hasParent <http://localhost:8080/fcrepo/rest/> ;
    ldp:contains <http://localhost:8080/fcrepo/rest/albums/ea/50/12/93/ea501293-64bf-430e-bf97-abcd64fda0c4> .
```

RDF and Non-RDF

The serializer has the option of including or excluding content based on whether it's an RDF resource or a non-RDF resource. Furthermore, those two types of content can be segregated into configurable directories.

- binaries are written to a filesystem path that corresponds to their relative repository path within the configured folder
- RDF is serialized to the configured format at a path that corresponds to their relative repository path within the configured folder but with a suitable file extension (.ttl for text/turtle)
- Server-managed triples are included in the serialized RDF
- the resource in the "describedby" header for binaries is serialized to the corresponding path of the binary, but within the RDF folder with a suitable file extension (.ttl for text/turtle)

Possible issues

- I was unable to test how resources whose paths contains characters that cannot be in filenames are serialized (try a resource with a ':' in the path on an OSX system for instance)
- bad stuff happens when your repository contains two resources with the same name (differentiated by \[1\])