Import and Export Tools

Note

This documentation relates to a work in progress during three focused sprints starting on August 29, 2016 and as such, the features and procedures described here may not be finalized and released.

There are two different methods available for exporting metadata and digital objects from a Fedora 4 repository. The first relies on Camel and the second is a command line utility. Both methods can export RDF and can be configured to also export binary resources.

Exporting Resources Using the Camel Serialization Module

The Camel Serializer Module is currently a proof of concept but has documentation and code. It works by listening for events about newly created resources, or edits to existing resources. As such, the re-indexer Camel component must be invoked to export resources that were created before it started listening, unless they are edited while the Camel Serializer is turned on.

Exporting Resources Using the Export Tool

The command line tool is part of the Import Export Utility. This tool can export an entire repository, or a subset, and is intended as a method for creating a complete backup of a single Fedora container and all its descendant resources. From there, it can be imported back into the same Fedora repository, a different Fedora Repository, or into another external system.

Installing the import-export utility

Fedora 4

The import/export utility is designed to work against the Fedora 4 REST API. Therefore, you first must have a running Fedora repository.

There are several ways to install a Fedora repository if you do not have one already available, see either:

1. Quick Start, or
2. Fedora Vagrant
    a. (optional step to turn off user authentication) edit the file at install_scripts/config and change `FEDORA_AUTH=true` to `FEDORA_AUTH=false` and save the file

Import/Export Utility

The import/export utility maybe be installed in one of two ways:

1. Download executable jar
2. Build from source

Preparing sample data

1. Go to Fedora in a browser. If you're using the Quick Start method, this will located at: http://localhost:8080/rest
2. Under the *Create New Child Resource* form on the right-hand side of the page, create a new resource called "albums"
3. Navigate to the new resource at http://localhost:8080/rest/albums
4. Under *type* on the right-hand side of the page, choose "binary" and select a file to upload from the resulting *Choose File* button
5. Leave the identifier field blank and click *Add*

You will now have a Container with a binary file ready to export at http://localhost:8080/rest/albums.

Using the export tool

Move to the directory where you downloaded the code. For Vagrant installs, this will be under /opt/fcrepo-import-export; otherwise, it will located in the directory where you cloned the GitHub repository when you build the tool from source. From there, you will need to locate the compiled jar file. If you build it from source, it will be in the GitHub repo at fcrepo-import-export/target/fcrepo-import-export-0.3.0-SNAPSHOT.jar.

To see all the available command line options:

```
java -jar fcrepo-import-export/target/fcrepo-import-export-0.3.0-SNAPSHOT.jar -h
```

Note: You will see "Error parsing args: Unrecognized option: -h" but this is bug and will be addressed in forthcoming releases.

To export all the RDF and binary resources in your sample container:

```
java -jar fcrepo-import-export/target/fcrepo-import-export-0.3.0-SNAPSHOT.jar --mode export --resource
http://localhost:8080/rest/albums --dir /tmp/test --binaries
```

You should now have RDF resources and binaries exported to /tmp/test.

Logging

To change the import-export logging level (default is INFO), set the fcrepo.log.importexport system property when running the command:

- Note, available logging levels are: TRACE, DEBUG, INFO, WARN, and ERROR

```
java -Dfcrepo.log.importexport=WARN -jar fcrepo-import-export/target/fcrepo-import-export-0.3.0-SNAPSHOT.jar --
mode export --resource http://localhost:8080/rest/albums --dir /tmp/test --binaries
```

Options

Exporting only RDF

If you do not want backups of the binaries, omit the --binaries option. You cannot export binaries without the rdf metadata description.

Authentication

Pass the -u option to provide a username and password for Fedora basic authentication.
If you are using the Vagrant installation, it will have Fedora Auth enabled and you will need to add the following to your command:

```
-u fedoraAdmin:secret3
```

Format Options

Pass -x option to use a different file extension on the exported rdf, or -l to change the rdf language used. For example if you want json-ld instead of turtle, use:

```
-x .json -l application/ld+json
```

The full list of available serializations is:

- application/ld+json
- application/n-triples
- application/rdf+xml
- text/n3 (or text/rdf+n3)
- text/plain
- text/turtle (or application/x-turtle)

(from RESTful HTTP API)

Importing resources

Exported resources may be imported back into the repository from which they came, or into the same path of another repository instance.

The following command will import the previous export into the same location of the same repository.

```
java -jar fcrepo-import-export/target/fcrepo-import-export-0.3.0-SNAPSHOT.jar --mode import --resource
http://localhost:8080/rest/albums --dir /tmp/test --binaries
```

To import into a different repository and/or path, use the map flag to ensure that the base URIs of the resources are properly translated in the new repository.

The following command will import the previous export into the same location of the same repository. In other words, the import would complete the migration of data from http://localhost:8080/rest/albums to http://localhost:8686/rest/albums.

```
java -jar fcrepo-import-export/target/fcrepo-import-export-0.3.0-SNAPSHOT.jar --mode import --resource
http://localhost:8686/rest/albums --dir /tmp/test --binaries --map http://localhost:8080/rest/albums,
http://localhost:8686/rest/albums
```