Islandora and Fedora 4;
The Atonement v3:
The Atonermenter
Project history and background
Fedora 4 Interest Group
Islandora Fedora 4 Interest Group

Convenors

- Nick Ruest (York University)
- Daniel Lamb (discoverygarden)

Terms of Reference

- The name of the iIG is the Islandora Fedora 4 Interest Group
- The purpose of the Islandora Fedora 4 Interest Group is to implement Fedora 4 in the context of Islandora.
  - Islandora/Fedora 4 Prospectus
  - Islandora/Fedora 4 Project Plan
- Specific goals, activities, outcomes may include:
  - Create a generic "upgradation" document for Islandora and:
    - Identify pilot partners
    - York University
  - Reviewing Fedora 3 to 4 Upgration Checklist and:
    - Identifying what features are necessary in Fedora 4
    - Mapping Fedora 3 features to Fedora 4 features
  - Mapping Fedora 4 features to Islandora
  - Determining how Islandora will leverage new Fedora 4 features
  - Outreach and communication
    - Engaging other interested Islandora community members
    - Recruiting developers to work on integration tasks
- The interest group will meet once a month virtually (e.g. via Skype). A call for agenda items will be posted to the Islandora Google Group 1 week prior to the meeting. The convenor will appoint a note taker for the meeting and meeting notes will be made available at some url.
- The convenors will produce a report to be submitted to the Islandora Roadmap Committee following the IF4iG’s meeting.

Links
Thank you to our sponsors:
Atonement
One Repo to rule them all, One Repo to find them, One Repo to bring them all and in the darkness bind them
github.com/Islandora-Labs/islandora
Documentation
About Islandora

Islandora is an open-source software framework designed to help institutions and organizations and their audiences collaboratively manage, and discover digital assets using a best-practices framework. Islandora was originally developed by the University of Prince Edward Island's Robertson Library, but is now implemented and contributed to by an ever-growing international community.

Islandora consists of:
- Sycx - Event driven middleware based on Apache Camel that synchronizes a Fedora 4 JCR with a Drupal CMS.
- Islandora - Fedora 4 Repository module
- Install - The is a development environment virtual machine for the Islandora and Fedora 4 project. It should work on any operating system that supports VirtualBox and Vagrant.

About this guide

The Technical Design documentation will help you:
- Understand the Islandora 7.x-2.x design rationale
- Importance of using an integration framework
- Using camel
- Inversion of control and camel
- Camel and scripting languages
- Islandora Sync
- Soar and Triple store indexing
- Islandora (Drupal)

The How to build documentation provides and overview on how the documentation is created, built, and deployed.

Installation

The installation section provides and overview on how to create a virtual development environment.

Contributing

If you would like to contribute, please get involved with the Islandora Fedora 4 Interest Group, and check out the contributing section. We love to hear from you!

If you would like to contribute code to the project, you will need to be covered by an Islandora Foundation Contributor License Agreement or Corporate Contributor License Agreement. Please see the Contributors page on islandora.ca for more information.

Sponsors

- Discoverygarden
MkDocs!
Contributing

http://islandora-labs.github.io/islandora/contributing/contributing
“All contributions are welcome: use-cases, documentation, code, patches, bug reports, feature requests, etc. You do not need to be a programmer to speak up!”
### Introduction

This is a development environment virtual machine for the Islandora and Fedora 4 project. It should work on any operating system that supports VirtualBox and Vagrant.
vagrant up

(that’s it!)
future.islandora.ca
Thank you University of Manitoba!
PCDM

Fedora, Hydra, Islandora… Repository communities unite!
Ontology for the Portland Common Data Model
Portland Common Data Model

<table>
<thead>
<tr>
<th>Namespace</th>
<th><a href="http://pcdm.org/models#">http://pcdm.org/models#</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Ontology for the Portland Common Data Model, intended to underlie a wide array of repository and DAMS applications.</td>
</tr>
<tr>
<td>Version</td>
<td>2015/09/28</td>
</tr>
<tr>
<td>Last Modified</td>
<td>2015-09-28</td>
</tr>
<tr>
<td>Prior version</td>
<td><a href="http://pcdm.org/2015/03/16/models">http://pcdm.org/2015/03/16/models</a></td>
</tr>
<tr>
<td>Published by</td>
<td><a href="http://www.duraspace.org/">http://www.duraspace.org/</a></td>
</tr>
<tr>
<td>See Also</td>
<td><a href="https://github.com/duraspace/pcdm/wiki">https://github.com/duraspace/pcdm/wiki</a></td>
</tr>
</tbody>
</table>

Table of Contents

Classes

Properties

Entity Definitions
WebAccessControl

...nobody loves XACML
WebAC Authorization Delegate

Fedora 4.x Documentation / ... / Authorization Delegates

Created by Mohamed Mohdseh Abdul Rashheed, last modified by Jared Whild on Oct 14, 2015

WebAC authorization Fedora module is an implementation of the still evolving draft by the W3C that proposes a decentralized authorization mechanism. See WebAccessControl specifications at the W3C website.

W3C's definition of WebAccessControl

WebAccessControl is a decentralized system for allowing different users and groups various forms of access to resources where users and groups are identified by HTTP URIs.

The WebAC module will enforce access control based on the Access Control List (ACL) RDF resource associated with the requested resource. In WebAC, an Access Control List (ACL) consists of a set of Authorizations. An Authorization is a single rule for access, such as "users alice and bob may write to resource foc", described with a set of RDF properties. Authorizations have the RDF type http://www.w3.org/ns/auth/acl#Authorization (for the remainder of this document, the http://www.w3.org/ns/auth/acl# namespace will be abbreviated with the prefix acl:).

The properties that may be used on an acl:Authorization are:

<table>
<thead>
<tr>
<th>Property</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl:accessTo</td>
<td>the URI of the protected resource</td>
</tr>
<tr>
<td>acl:agent</td>
<td>the user</td>
</tr>
<tr>
<td>acl:mode</td>
<td>the type of access (WebAC defines several modes: acl:Read, acl:Write, acl:Append, and acl:Control)</td>
</tr>
<tr>
<td>acl:accessToClass</td>
<td>an RDF class of protected resources (N.B., not implemented in the first version of this module)</td>
</tr>
<tr>
<td>acl:agentClass</td>
<td>an RDF class of users (N.B., not implemented in the first version of this module)</td>
</tr>
</tbody>
</table>
Upgration!

How are we going to get there?

...or portmanteaus are awesome!
York University

Solution Packs:
Collection, Audio, Book, Compound, Large Image, Video, Web Archive
Project Overview

The York University Libraries upgration project identifies collections that cover the range of object models that the repository uses. The conservative goal is to perform an upgration on the collections listed below. The stretch goal is an upgration all of all objects in the repository.

By upgration, we mean upgrading and migrating objects and datastreams, along with security restrictions (XACML), in Fedora 3.6.0 to Fedora 4.x. Moreover, we will develop a strategy for upgrading and migrating our content models, including inline XML datastreams, managed datastreams, and external datastreams.

York University Digital Library (YUDL) is an Islandora repository that run on the HEAD version of all Islandora Foundation modules. The repository is run as close a stock/generic Islandora instance where possible. Therefore, this upgration pilot can serve as a basis for a generic Islandora Fedora 3.x to Fedora 4.x upgriation.

- Collection Description(s)
- Object Models
- Fedora 3 Details
  - Storage: Legacy storage (or Akubra)
  - XML metadata: datastreams
  - XML metadata: inline
  - Content models
  - Datastream types (inline, managed, redacted, and external)
- Identifiers
- Indexing strategies (GSRepair, RI-Search vs. F4 approaches)
- Replication/Journaling
- Security policies: XACML
- OAIPMH
- Versions
- Disseminators
- Audit history
- Fedora 4 Details

Collection Description(s)

York University Digital Library contains approximately 200,000 unique digital assets.

Jean-Augustine Fonds
Property Mappings
fcrepo3->fcrepo4

Object properties
## 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>dcterm:identifier</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>ownerld</td>
<td>fedora3model:ownerld†</td>
<td>nruest</td>
</tr>
</tbody>
</table>

† The `fedora3model` namespace is not a published namespace. It is a representation of the fcrepo3 namespace `info:fedora/fedora-system:def/model`.
‡ Not yet implemented
fcrepo3->fcrepo4

Datastream properties
<table>
<thead>
<tr>
<th>fcrepo3</th>
<th>fcrepo4</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<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>
<tr>
<td>Checksum</td>
<td>cryptofunc:hasalgorithm‡</td>
<td></td>
</tr>
</tbody>
</table>

† The fedora3model namespace is not a published namespace. It is a representation of the fcrepo3 namespace info:fedora/fedora-system:ndef/model1.
‡ Not yet implemented
fcrepo3->fcrepo4

RELS-EXT/RELS-INT
Fedora 3.x namespace RELS-EXT predicates

$ grep -R "FEDORA_RELS_EXT_URI" Islandora-7.x

Islandora
- isMemberOfCollection
- isMemberOf

Image Annotation
- isAnnotationOf

Compound
- is ConstituentOf
- is PartOf

Islandora namespace RELS-EXT predicates

$ grep -R "ISLANDORA_RELS_EXT_URI" Islandora-7.x

Book
- isPageOf
- isSequenceNumber
- isPageNumber
- isSection

Image Annotation
- targetedBy
- targets
- hasColor
- hasURN
- strokeWidth
- isEntity
- isAnnotationType

OCR
Islandora Ontology

http://islandora.ca/ontology/relsext/#
http://islandora.ca/ontology/relrint/#
Islandora RELS-EXT Ontology

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Namespace</td>
<td><a href="http://islandora.ca/ontology/relesext#">http://islandora.ca/ontology/relesext#</a></td>
</tr>
<tr>
<td>Description</td>
<td>Ontology for the Islandora RELS-EXT data model, intended primarily to make it possible to expose Islandora-curated RDF predicates via de-reference-able URIs.</td>
</tr>
<tr>
<td>Version</td>
<td>v1/2015/05/11</td>
</tr>
<tr>
<td>Last Modified</td>
<td>2015-05-11</td>
</tr>
<tr>
<td>Published by</td>
<td><a href="http://islandora.ca/">http://islandora.ca/</a></td>
</tr>
</tbody>
</table>

### Table of Contents

#### Properties

dateIssued generate_ocr hasColor hasLanguage hasURN isAnnotationType isEntity isManageableByRole isManageableByUser isPageNumber isPageOf isSection isSequenceNumber isSequenceNumberOf$escaped_pid isViewableByRole isViewableByUser targetBy targets

#### Entity Definitions

##### Properties
Islandora RELS-INT Ontology

<table>
<thead>
<tr>
<th>Namespace</th>
<th><a href="http://islandora.ca/ontology/relsint#">http://islandora.ca/ontology/relsint#</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Ontology for the Islandora RELS-INT data model, intended primarily to make it possible to expose Islandora-curated RDF predicates via de-reference-able URIs.</td>
</tr>
<tr>
<td>Version</td>
<td>v1/2015/05/11</td>
</tr>
<tr>
<td>Last Modified</td>
<td>2015-05-11</td>
</tr>
<tr>
<td>Published by</td>
<td><a href="http://islandora.ca/">http://islandora.ca/</a></td>
</tr>
</tbody>
</table>

Table of Contents

Properties

height width

Entity Definitions

Properties
Islandora Ontology

https://github.com/Islandora-Labs/islandora_ontology
fcrepo3->fcrepo4

auditTrail
## Audit log migration

### auditTrail mapping

<table>
<thead>
<tr>
<th>fc:rep03 event</th>
<th>fc:rep04 Event Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>addDatastream</td>
<td>premis:ing‡</td>
</tr>
<tr>
<td>modifyDataStreamByReference</td>
<td>audit:contentModification/metadataModification‡</td>
</tr>
<tr>
<td>modifyObject</td>
<td>audit:resourceModification‡</td>
</tr>
<tr>
<td>modifyObject (checksum validation)</td>
<td>premis:validation‡</td>
</tr>
<tr>
<td>modifyDataStreamByValue</td>
<td>audit:contentModification/metadataModification‡</td>
</tr>
<tr>
<td>purgeDatastream</td>
<td>audit:contentRemoval‡</td>
</tr>
</tbody>
</table>

† The `fc:rep03model` namespace is not a published namespace. It is a representation of the fc:rep03 namespace `info:fedora/fedora-system:def/model`.
‡ Not yet implemented
migration-utils

https://github.com/fcrepo4-labs/migration-utils
7.x-2.x
Design Goals

- Utilize Fedora 4 to its fullest
- Easier to install
- Easier to develop and contribute
- Easier to use
  - Enhanced user experience
  - Faster searching, display, and ingest
  - Expose more control of your repository through UI
- Easier to scale
How Do We Achieve These Goals?

- Installation bash scripts
- Vagrant development environment
- Tight Drupal integration
- Asynchronous interactions (when possible)
- Transformation logic in the UI
- Decoupling
  - Decoupling… Decoupling… Decoupling… Decoupling…
And How Do We Achieve THOSE Goals?

Apache Camel
Why Camel?

- Islandora is middleware!
  - Camel is a framework for building middleware
    - Provides workflow-like structure for the code
  - It’s already solved a lot of really hard problems
    - Message splitting, transformations, concurrent processing, error handling / redelivery…
  - Works out of the box with everything you can think of
    - Filesystems, emails, web services, databases, and now… Fedora (thanks Aaron Coburn!)
What is Camel’s Role in Islandora?

● Dirty Work
  ○ Data transformation
  ○ Calling out to command line tools
  ○ Interacting with other APIs

● Doing the Dirty Work Sanely
  ○ Transactions, redelivery, dead letter channels

● Man in the Middle
  ○ Fedora and Drupal never talk to each other
What does a system with Camel at its core look like?
Islandora Sync

- Event driven middleware that synchronizes Fedora resources with Drupal nodes
- Does 3 things
  - Deletes things that no longer exist
  - Upserts everything else
    - Transforms ld+json RDF from Fedora into Drupal node JSON and issues PUT requests.
  - Generates derivatives
Islandora Services

- Micro-services invoked in response to Drupal hooks
- Wraps the Fedora 4 REST API
  - Accepts data in Drupal’s formats
  - Constructs PCDM-compliant resources in Fedora
    - Transforms Drupal node JSON into SPARQL Update queries
- Doesn’t wait around (lets Sync do the time consuming stuff)
Islandora Commands

- Command line PHP tools
- Based on Symfony components
- Utilized by the Camel layer for when you need advanced processing but don’t want to dip into Java
- Replaces the use of Drush for when you need to execute non-Drupal code
Islandora Camel Component

- Executes Islandora Commands
- Saves on lots of boilerplate
- Bubbles up errors and exceptions in PHP and throws them in Camel
Benefits of This Approach

● Better Drupal Experience
  ○ Content is Drupal nodes
    ■ Developers work with Drupal content, not Fedora content
    ■ Take full advantage of drupal hooks and theming
  ○ UI controlled content modeling (fields)
  ○ UI controlled display (site building)
  ○ Views!
Benefits of This Approach (Cont’d)

- 3rd Party Modules
  - apachesolr
  - rdfx
  - services
  - xml_field
  - xpath_field
  - field_permission
  - and many many more….
Benefits of This Approach (Cont’d)

- RDF / Field synchronization
- RDFA enriched output
- All exposed through Drupal UI
Benefits of This Approach (Cont’d)

- XML Fields
- XPath Fields
- Extract metadata from XML and map to RDF
- Also exposed through Drupal UI
Benefits of This Approach (Cont’d)

MODS XML

```xml
<?xml version="1.0"?>
<titleInfo>
<title>Oh, you’re an engineer?</title>
<subTitle>Please continue telling me why my career is a waste of time</subTitle>
</titleInfo>
<nameInfo>
<name type="personal">
<namePart>The Internet</namePart>
</name>
</nameInfo>
```

A MODS record for the Fedora resource

MODS Title (Derived with /mods/titleInfo/title)

Oh, you're an engineer?

```
foaf: name
admin

modsrdf: title
Oh, you're an engineer?

nfo: uuid
2bc4d9af-7ae7-440e-a7f7-d7b59adb32ca
```
Benefits of This Approach (Cont’d)

- Now using drupal.org solr modules
- No more GSearch
  - No more XSLT's
- Maintaining much less code

Related Projects

Projects that are co-maintained in combination with this one

**Apache Solr Attachments** (7.x & 6.x)
Send your files to Apache Tika and allow Drupal to search in your attachments/files

**Apache Solr Multisite search** (7.x & 6.x)
Combine any combination of 6.x-3.x and 7.x-1.x Drupal sites to let them all search and index to the same Apache Solr core index

**Apache Solr Devel** (7.x-1.x)
Allows to fetch your entity from the solr index in a similar way as you can show the entity object in the devel tab of any entity. Very useful to find out what values you have in Solr for each field.

**Rich Snippets** (7.x-1.x)
By using RDF and schema.org semantic data for your structured content you can now show similar snippets as the ones Google provides

**Apache Solr Sort** (7.x-1.x)
Adds support for the Solr Grouping feature and adds an UI to enable/disable sort fields

**Apache Solr API Examples** (7.x-1.x, 6.x-3.x)
Shows you, with real examples, how to program Drupal modules using the Apache Solr Search Integration API

**Apache Solr Parallel Indexing** (7.x-1.x, 6.x-3.x)
Using PHP socket streams we can execute a number of php processes at the same time to make indexing speed faster. Speed increase is limited by the number of CPU's
Current Functionality?
Next Steps
Questions/Discussion