Islandora and Fedora 4; The Atonement.
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 llG 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 "upgration" 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 (eg. 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
Documentation
About Islandora

Islandora is an open-source software framework designed to help institutions and organizations and their audiences collaborate 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:

- **Sync** - 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](https://islandora.ca/about) 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
- Solr and Triple store indexing
- Islandora (Drupal)

The [How to build documentation](https://islandora.ca/about) provides an overview on how the documentation is created, built, and deployed.

Installation

The [installation section](https://islandora.ca/about) provides an 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](https://islandora.ca/about) or [Corporate Contributor License Agreement](https://islandora.ca/about). Please see the [Contributors page](https://islandora.ca/about) on islandora.ca for more information.

Sponsors

- Discoverygarden
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!”
DevOps
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
PCDM

Fedora, Hydra, Islandora… Repository communities unite!
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 would 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

- Web 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
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"?>
      xsi:schemaLocation="http://www.loc.gov/mods/v3 mods.xsd">
  <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>
</mods>
```

A MODS record for the Fedora resource.

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

```
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
Next Steps
Questions/Discussion