APPENDIX E - Glossary

Collection

Collections are a way of grouping related objects together in Fedora. They function much like directories on a computer; a collection can "contain" any number of related objects and sub-collections. For more technical detail, see **Collection Object**.

Collection Object

Everything stored in Fedora is an object; including collections. Collection objects provide a means to organize other objects via an *isMemberOf* element in an **RDF** statement.

Collection Policy

The collection policy is a **Datastream** that specifies which **content models** are associated with a particular **collection object**. This determines which file types can be ingested into the collection. The **Datastream ID** (DSID) for a Collection Policy will always be COLLECTION_POLICY.

Composite Model (DS-COMPOSITE-MODEL)

The "DS-COMPOSITE-MODEL" defines what data streams comprise an object, whether or not these datastreams are optional or enforced, and what mime type the datastream can, should, or must contain.

Content Model

Content models describe the characteristics of objects in the Fedora repository. Islandora extends Fedora's content model architecture with a datastream that has the Datastream ID (DSID) ISLANDORA_CM. The "Content Model" is a is a legacy idea, and all "Content Models" should be (if they haven't already been) moved to the "DS-COMPOSITE-MODEL" datastream.

Datastream

Datastreams are elements of **objects** that represent content. They may contain metadata about the object. Each object has a number of default datastreams, notably a Dublin Core Datastream and a RELS-EXT (Fedora Object-to-Object Relationship Data Datastream. Datastreams can also contain other useful information, and different types of objects will have different datasteams; for example, a **collection object** has a **collection policy** Datastream, while an image might contain a thumbnail Datastream.

Datastream ID (DSID)

The Datastream ID is a unique identifier within the scope of the digital object. So, for example, an object can only have one Datastream with a DSID of COLLECTION_VIEW, but any other object can also have a Datastream with the same DSID. See APPENDIX C: Datastream Reference for more information.

Datastream Label

The Datastream label is a human-readable title given to a Datastream. It does not need to be unique, but it should be informative enough for a user to understand the purpose of the Datastream. When you 'view' a Datastream in the Islandora interface, you will see the Datastream ID of the object.

Drupal (Content Management System)

Drupal is an open source content management system (CMS) written in PHP. Known for being extremely flexible and extensible, Drupal is supported by a community of over 630,000 users and developers. Drupal sites can be customized and themed in a wide variety of ways.

Drupal functions as the "front-end" of Islandora.

Drupal Roles

Roles are a way of assigning specific permissions to a group of users. Any user assigned to a role will have the same permissions as all other users assigned to that role. This allows you to control which users have permission to view, edit, or delete content in both Drupal and the Fedora repository.[1]

Dublin Core

Dublin Core is a metadata standard that provides a small set of text elements through which most resources can be described and catalogued.

Fedora Commons (Repository Software)

Fedora (Flexible Extensible Digital Object Repository Architecture) is an open source architecture for storing, managing, and accessing digital content in the form of *digital objects*. Fedora defines a set of abstractions for expressing digital objects, asserting relationships among digital objects, and linking "behaviors" (i.e., services) to digital objects.

The Fedora repository functions as the "back-end" of Islandora; files are stores as objects in the repository and accessed through the "front-end" (i.e., Drupal).

Faceted Search

Faceted searching allows users to filter a set of search results using predetermined classifications. For example, a user might search by title and then have the option to filter those results by selecting from a list of subjects and collections.

You can use Solr to configure your faceted search options.

FOXML

FOXML (Fedora Object Extensible Markup Language) is a simple XML format that directly expresses the Fedora Digital Object Model. Digital objects are stored internally in a Fedora repository in the FOXML format. In addition, FOXML can be used for ingesting and exporting objects to and from Fedora repositories.

Any Fedora object is represented by a complete FOXML document. For more information, please see Managing access control with permissions and user roles.

Ingest

To ingest an object is to add an entry for it in the Fedora Repository. This is done primarily through the Islandora Repository interface in an Islandora site's Drupal front-end. Upon ingest, an object is created in Fedora, and its associated PID and datastreams are attached to it.

The word 'ingest' is used in repository language because repository software, rather than having an object and associated components forced upon it, takes in and processes the object itself. Often, when referring to the process of ingesting objects, Islandora and its associated documentation use other terms such as 'import' or 'add object'. In such contexts, these terms generally refer to the ingest process.

Namespace

The namespace is a component of the **persistent identifier** (namespace:ID). The namespace is defined in the **collection policy**, and will then be used to generate PIDs for the objects ingested into that collection.

Object (Also "Fedora Object")

Everything stored in a Fedora repository is treated as an object. Objects are made up of three components: a **Persistent Identifier** (PID), System Properties, and one or more **Datastreams**. The generic nature of Fedora's digital object model allows a wide variety of digital content to be stored and delivered to users.

Open Source

Open source describes a method of software development that promotes access to the end product's source code. Islandora is an open source product with an active development community, operating under the GPL license (3.0).

Parent-Type Object

A Parent-Type Object is an object that has a relationship with the islandora:collectionCModel declared in the policy datastream, making it possible for additional objects to have a relationship with (or be "children" of) that object.

Persistent Identifier (PID)

Every **object** in a Fedora repository has a unique persistent identifier of the format **Namespace**:ID. This identifier is assigned when the object is created, and cannot be changed. The PID can be user-generated or managed based on information specified in the **collection policy**.

Relationship (Also "RDF")

Objects in a Fedora repository are organized using Resource Description Framework (RDF) statements.These statements define the relationships between objects in the following format:

<subjectFedoraObject> <relationshipProperty> <targetFedoraObject>

These RDF statements can define a number of relationships; for example:

<imageOfMap> <isMemberOf> <myMapCollection>

In this example, the map image belongs to the map collection.

Root Collection

The root collection is the top-level collection in the repository. Out-of-the-box, the root collection has the PID *islandora:root*.

Viewer

A "Viewer" is an add-on package that allows a solution pack to embed, display, or playback a particular object in a web accessible format. Viewers are typically a combination of a Drupal Library and a Drupal Module. The Drupal Library is the package of code that represents the player or display mechanism, and the Drupal Module is the code package that allows the Drupal Library to be accessed from within the Drupal environment.

Virtual Machine Image

The Virtual Machine Image allows you to mount a fully working version of Islandora on your local machine as a separate virtual machine.

XML Schema

XML schemas are used to validate XML documents. The XML document is compared to a particular schema in order to test its validity in a specific context. In Islandora the metadata schemas are frequently used by XML Forms to create and validate ingest forms.

XSD

And XSD file - which stands for XML Schema Document - contains a set of rules to which an XML document must conform in order to be considered 'valid'.

In the Form Builder context, the XSD file defines the schema that you are creating a form for. Schema's XSD files are usually hosted by the project's website, although you may upload a schema directly to your server.