Getting Started with Islandora

This chapter focuses on explaining the way that Islandora understands content, and providing basic instructions for how you can create, view, update, and delete (purge) content in your repository. This chapter presumes that you have access to a running instance of Islandora, either through the online sandbox or a virtual machine image on your local machine. Both the online sandbox and the VM come with pre-installed collections that will be used as examples in this documentation.

If you do not have access to an instance of Islandora, please review the Accessing our Sandbox section.

How Islandora Understands Content

Before accessing and using Islandora, it is important to learn how Islandora understands your content. Islandora uses Fedora Commons' flexible digital object model, relationships, and content model architecture. If you are familiar with these concepts, you may find this section repetitive, because it borrows heavily from the Fedora Commons documentation to provide what is most relevant for Islandora.

Understanding Islandora means understanding the following:

- Everything in Islandora's repository is an **Object**
- Objects are made up of **Datastreams**
- Objects have **Relationships** to one another
- Objects have **Persistent Identifiers (PIDs)** that are unique in your repository

Everything in Islandora's repository is an object

Everything in Islandora's Fedora Commons Repository is an *object* made up of *datastreams*. An object is like a directory or folder full of different files (the datastreams). However, unlike the folder system on your computer (which contain other folders), Islandora's information architecture is graph-like, more like a network. There are several kinds of objects, and each object performs a different function. Islandora's main object types are **Content Model Objects**, **Collection Objects**, and **Data Objects**.

Content Model Object

A Content Model Object is a "template" for a particular type of content. Content model objects are usually built through a Solution Pack. Other analogies are a cookie-cutter or recipe ingredient list. Fedora Commons has its own notion of a Content Model, which you can read more about here. This version of Islandora uses the Fedora Commons 3.x content model architecture.

Unlike previous versions of Islandora (6.x), Islandora 7's content model is very lightweight and defined in a DS-COMPOSITE-MODEL datastream. The DS-COMPOSITE-MODEL defines what types of files can be added to an object, including whether these datastreams are optional or required.

Collection Object

Collection objects are Fedora objects that have "child" objects that are gathered into a collection in the Islandora display. If you want to make a collection of PDFs in Islandora, you would start by creating a collection object that has the PDF Content Model. Because of Fedora's ability to automatically create relationships between objects, every data object you add to this collection becomes a member of that collection with a content model of PDF.

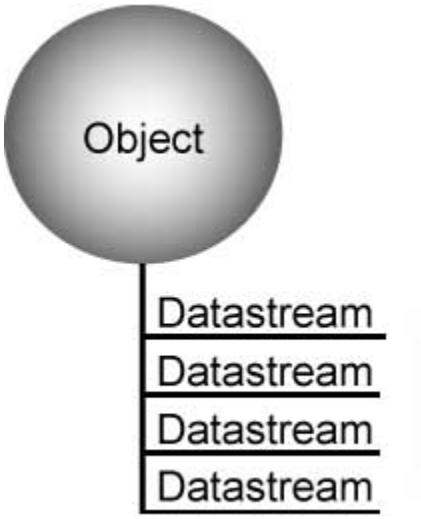
Every instance of Islandora starts with a root collection object. This collection object has the Collection Content Model, enabling more collections to be created as children (sub-collections) of the root collection. These sub-collections can have their own sub-collections, or can contain any type of data object as defined in the collection policy datastream of each collection.

Data Object

Data objects are the files ingested into the repository and any associated metadata, derivatives, or related files that should be managed as a single digital asset in the repository. For example, one image (such as a TIFF file) and its associated datastreams (such as descriptive and technical metadata, thumbnails, and access derivatives in other formats) make up a "data object." Fedora also stores relationship metadata in datastreams that connect objects to parent collections, content models, and other related objects.

Objects have datastreams

Each object has datastreams, which are the parts of an object.



Datastreams may include content files (such as a PDF or an image), metadata, or information about relationships to content models or other objects. Every object in Islandora has at least three datastreams in common (DC, RELS-EXT, and TN) as well as specialized datastreams to serve the functions of that object. The relationship of an object to other objects is stored in the RELS-EXT datastream, simple metadata about the object is stored in the Dublin Core (DC) datastream, and a display thumbnail for the object is stored in the TN datastream.

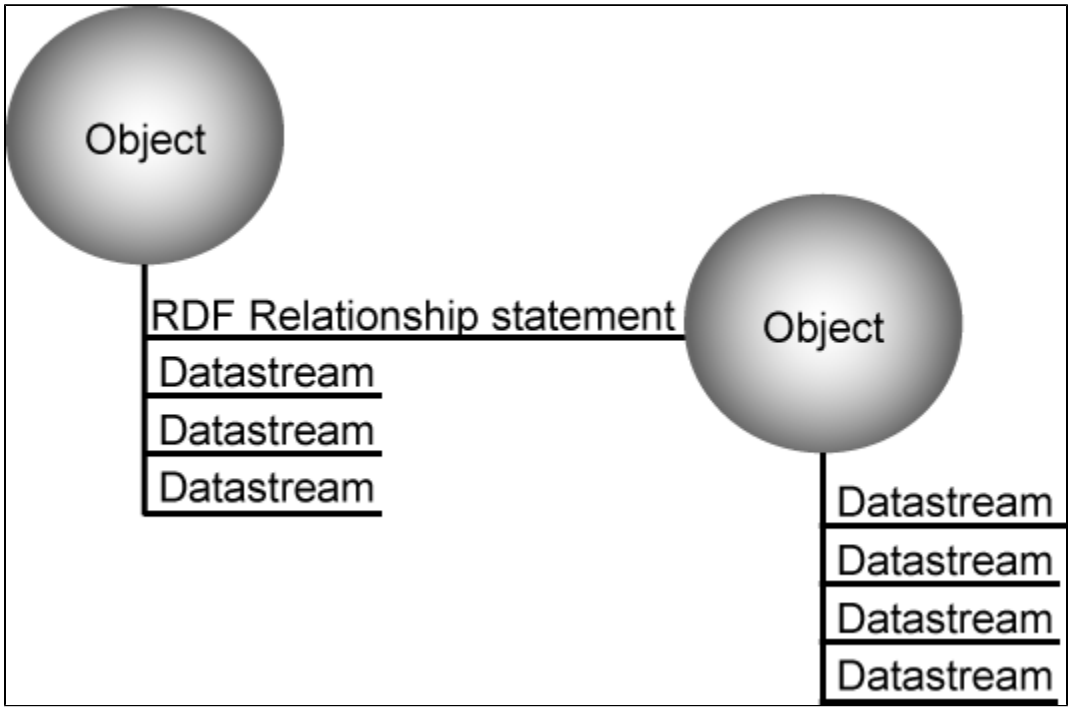
For a complete list of datastreams in Islandora, see APPENDIX C - Datastream Reference.

Objects have states

Every object in Islandora can also be set to one or more "states" in Fedora. Fedora states include "Active," "Inactive," or "Deleted." These states are used in different ways by Islandora code. By default, only "Active" objects display to anonymous users. Inactive states can be used for objects that aren't ready to be published (for example, objects waiting for approval in an ingest workflow). Objects can also be set to a state of "Deleted," which maintains a copy of the object in the repository but restricts viewing and management to certain roles. This is in contrast to purging an object, which permanently removes the object and its administrative history from the repository.

Objects have relationships

Objects in Islandora have relationships to one another. These relationships are stored in a RELS-EXT datastream in an object, which usually has the label “Fedora Object-to-Object Relationship Metadata.” This datastream is written in RDF/XML.

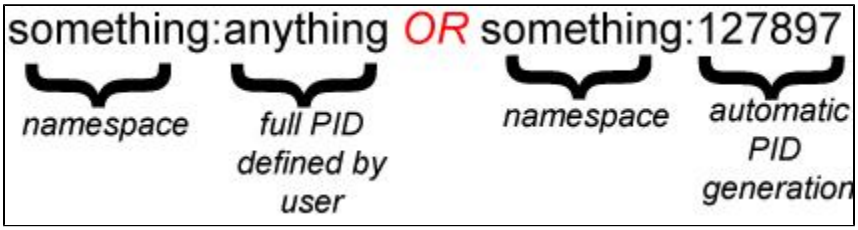


The RDF/XML statements in this datastream will indicate, for example, what collection object(s) an object belongs to, what content model object it subscribes to, and what its persistent identifier (PID) is.

A default set of common relationships is defined in the Fedora Relationship Ontology. More information on digital object relationships is available from the Digital Object Relationships section of the Fedora Commons documentation. References to these documents are also provided in the selected reading section of this guide.

Objects have unique, persistent identifiers

Each object in Islandora has a unique identifier called a PID (Persistent IDentifier). No two PIDs in your repository can be the same. PIDs look like this:



In the first case, the entire PID has been specified by a user. In the second case, the first part of the PID (the **namespace prefix**) has been added by the user, and the system is automatically assigning a number to make up the second part of the PID. Islandora supports the creation of PIDs based on UUIDs as well.

PID namespace prefixes do not need to be meaningful, they just need to be unique. However, making these namespaces meaningful can help you sort and find items in your repository. For example, a namespace might refer to the type of content (such as audio:) or it may refer to the name of an institution or collection (upei: OR stmatthews:). As you navigate Islandora, you will see PID namespaces mentioned frequently, and you will be prompted to choose PID namespace prefixes or PID namespaces.

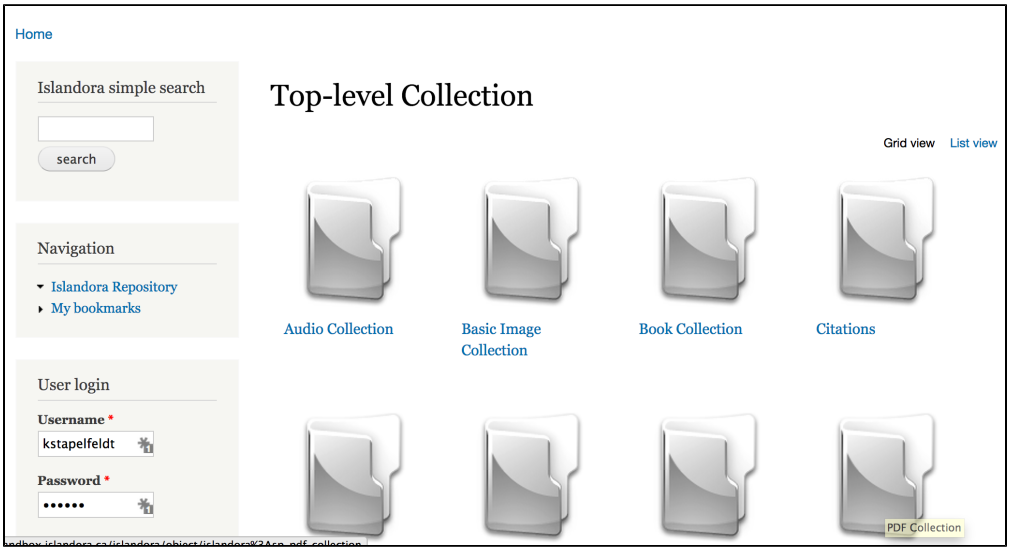
Another way of understanding the PID is as the basis for your repository's Uniform Resource Identifiers (URIs) which uniquely identify items in your repository. To learn more about PIDs and PID namespace prefixes, visit Fedora Commons documentation; specifically, the section on Fedora Identifiers.

Managing and Building Collections in Islandora

Once you have logged into your Islandora site as an administrator, you can view a list of all current collections by clicking the **Islandora Repository** link.

You should see any collections that are installed as part of a **Solution Pack,** and they will appear with a default folder icon.

These tutorials assume that you have installed one or more Solution Packs. Please review the milestone 5 - Installing the Islandora Essential Modules and the Installing the Islandora Enhancement Modules if you are not sure if you have Solution Packs installed. If no solution packs are installed, you will not be able to create collections of content in Islandora.



The following tutorials will guide you through the basics of managing and building your Islandora collections: